

IAP20 Rec'd PCT/PTO 26 JAN 2006

明 細 書

発明の名称

画像処理装置、画像処理方法及び歪補正方法

技術分野

本発明は、主に、デジタルカメラ等の電子的撮像装置に用いられる画像処理装置及び画像処理方法に関し、特に、回路規模やデータ転送量を増大させることなく、歪補正機能を実現し、小領域（例えばブロックライン）単位での歪補正処理を行う場合に、小領域の空間的な位置を算出可能とした画像処理装置、画像処理方法及び歪補正方法に関する。

背景技術

デジタルカメラ、銀塩カメラを問わず、一般的にカメラの光学系においては歪曲収差が現れる。また、現在発売されているカメラでは、光学ズームが可能な機種が殆どであり、その場合、ワイド端からテレ端にかけて、歪曲収差の状態が変化する。ワイド端では樽型歪が多く、テレ端では糸巻き型歪が多い。

歪曲収差は、例えば格子状の被写体を撮影すると、樽型歪、糸巻き型歪として観測される。

図5 6は格子状の被写体、図5 7は樽型歪を生じた撮影画像、図5 8は糸巻き型歪を生じた撮影画像である。

ところで、デジタルカメラにおいては、CCD等の撮像素子のデータに対し、様々な画像処理を行なった後、JPEG等の圧縮方式で圧縮したデータをメモリカード等の記録媒体に記録する。

図5 9は一般的なデジタルカメラで行われている画像処理手順の概念を示している。CCDにより取り込まれた撮像信号にプリプロセスで画素欠陥処理、A/D変換等を行い、得られた画像データをSDRAM等のフレームメモリに一旦記憶する。次に、フレームメモリから読み出された画像データに対してイメージプ

ロセスにより各種の画像処理を行い、さらにJ P E G処理で画像圧縮して記録媒体としてのメモリカード等に記録する。

図60は従来のデジタルカメラの画像処理装置のブロック図（例えば特開2000-312327号公報での従来例）を示している。

図60において、従来の画像処理装置は、バス103にCPU104と共にプリプロセス回路102、複数のイメージプロセス回路106-1～106-n、J P E G処理部107、フレームメモリ105及び記録媒体としてのメモリカード108が接続されている。そして、CPU104の制御により、CCD101からの撮像信号がプリプロセス回路102で画素欠陥処理やA/D変換等が施された後、バス103を通してフレームメモリ105に一旦記憶される。次に、フレームメモリ105から画像データを読み出し、バス103を通してイメージプロセス回路106-1に入力して所定の画像処理を行い、再びバス103を通してフレームメモリ105に書き直す。以下同様にして、バス103を介してフレームメモリ105とイメージプロセス回路106-2～106-nとの間でデータのやり取りを順次行って、最後にJ P E G処理部107でJ P E G圧縮処理を行い、画像処理したデータをフレームメモリ105に一旦記憶し、フレームメモリ105から読み出した処理データをメモリカード等108に記録するようになっている。以上の各段のイメージプロセス及びJ P E G処理では、小領域（ブロックライン）を単位として画像処理が行われる。

ところで、イメージプロセス回路の少なくとも1つにおいて図56～図58で述べたような歪補正処理を行う方法として、下記の光学的に表現される歪補正式〔式5〕を用いることによって撮像画像の歪曲収差を補正することが行われる。

〔式5〕

$$Z^2 = (X - X_d)^2 + (Y - Y_d)^2$$

$$\begin{cases} X' = (X - X_d) \cdot [1 + A \cdot Z^2] + X_d \\ Y' = (Y - Y_d) \cdot [1 + A \cdot Z^2] + Y_d \end{cases}$$

(X, Y)	: 補正画像内での座標
(X', Y')	: 撮影画像内での座標
(X _d , Y _d)	: 歪中心位置
A	: 歪補正係数

しかしながら、光学的に表現される上記の〔式5〕を用いて歪補正を行った場合、補正画像に過不足が生じる。

図61、図62は樽型歪、糸巻き型歪を生じた撮影画像を、上記の〔式5〕によって補正する動作を説明するものである。

図61は点線のように樽型に歪んだ元画像（図示左）を〔式5〕にて補正した場合であって、補正された画像（図示右）は画像出力範囲からはみ出して無駄になる部分を生じている。

図62は点線のように糸巻き型に歪んだ元画像（図示左）を〔式5〕にて補正した場合であって、補正された画像（図示右）は画像出力範囲より小さくなりデータが不足する部分が生じる。

このような図61、図62の不具合に対して、先行出願である日本国特開平9-098340号公報では、歪補正後に、電子ズームを行うことで補正により伸長した分を縮小して元の位置まで戻して、歪補正により失われる画像情報を最小限に抑えている。

一方、通常、CCD等の撮像素子の読み出しは、(1)本撮影用の全画素読み出し、(2)スルー画用の間引き読み出し、の2種類の読み出しが可能である。間引き読み出し等によって、撮像したデータから一部を取り込んだ画像データは、アスペクト比が1:1ではないため、そのままでは正確な歪補正が行えない。

また、従来、この画像処理の一部として歪補正を行う先行技術が開示されている。例えば、特開平6-181530号公報、特開平10-224695号公報がある。

特開平6-181530号公報には、撮影時における撮像ズームレンズの撮像ポジションが歪曲収差の大きいポジション内であることが検出手段にて検出されている場合には、この撮像ズームレンズによって生じた像の幾何学的歪みを、固体撮像素子の撮像データを幾何学的変形に基づき読み出すことにより補正することが記載されている。

特開平10-224695号公報には、光学系を経由した光を受光する固体撮

像デバイスで撮像したデータをランダムアクセスビデオメモリに記憶し、ランダム読み出しタイミング発生回路は光学系によって生ずる収差を補正するための収差補正データを有し、この収差補正データに基づいて固体撮像デバイスによって生成された信号を所定の順に読み出してビデオ信号を生成することで、光学系の歪曲収差を補正することが記載されている。

しかしながら、特開2000-312327号公報では、フレームメモリと各イメージプロセス回路間でデータのやり取りが多く、バスのデータ転送量が増大する。さらに、小領域（ブロックライン）単位での歪補正処理については全く記載されていない。

また、特開平9-098340号公報では、樽型歪の補正にしか言及しておらず、撮像中心に一番近い点を基準にしており、原理的に樽型歪の補正のみである。

糸巻き歪に対しては、補正された画像が画像出力範囲より小さくなり画像データが不足する部分が生じる。さらに、図22に示すような陣笠型歪を補正するには、歪補正式に4次項以上が必要になるが、元画像データを最大限使うための補正倍率Mを解析的に求めることが困難であった。

特開平6-181530号公報では、歪補正処理を行うことが述べられているが、図36に示すようにCCDの撮像面における画素（黒丸印にて示される縦横の格子点）31に対して、本来直線であるべき被写体像が光学レンズの歪曲収差により湾曲して結像（符号32にて示す）すると、歪曲収差補正回路内のラインメモリに大容量バッファメモリが必要となる。また、バッファメモリの容量で処理できる画像サイズが制限されてしまう。さらに、バッファメモリに保持されているデータの空間的な位置の確定方法等についての記載がない。さらに、特開平6-181530号公報では、小領域（ブロックライン）単位での歪補正処理については全く記載されていない。

また、特開10-224695号公報では、ランダムアクセスで処理するのに、バッファメモリは必要ないが、一般的にSDRAM等のメモリをランダムアクセスすると、転送時間がかかる。

そこで、本発明の第1の目的は、撮像した元データを有効に活用して歪補正処理を行い、補正した画像が画像出力範囲に対して無駄なく有効に出力でき、糸巻き型歪、樽型歪、及び陣笠型歪にも対応できる歪補正処理が可能な画像処理装置及び画像処理方法を提供することである。

本発明の第2の目的は、小領域（例えばブロックライン）の空間的な位置を算出可能とし、小領域単位での歪補正処理を実現可能とした画像処理装置及び歪補正方法を提供することである。

本発明の第3の目的は、間引き取り込みされた画像データに対しても、正確な歪補正を行える一方、デジタルズームなど、任意領域を切り出したデータに対しても正確な歪補正を行える画像処理装置及び歪補正方法を提供することである。

本発明の第4の目的は、バスの転送量やメモリの容量を大きく増大させることなく、歪補正処理を実現できる画像処理装置及び画像処理方法を提供することである。

発明の開示

本発明による画像処理装置は、歪補正手段を有する画像処理装置であって、該画像処理装置は、さらに、前記歪補正手段が歪補正処理を行う入力画像範囲を算出する歪補正範囲算出部を有する。

このように、歪補正処理を行う入力画像範囲を算出する歪補正範囲算出部を設けることにより、出力画像（即ち、歪補正後の画像）の出力範囲の一部または全部に相当する領域が入力画像（即ち、撮像画像）内に占める範囲を算出することで、歪補正によって得られる補正画像は出力すべき出力範囲に対して過不足なく出力することが可能となり、出力範囲に対して補正画像がはみ出したり不足したりすることのないようにできる。

好ましくは、前記歪補正範囲算出部は、補間座標を生成する座標生成部と、前記生成された補間座標に対して所定の歪補正式を適用して変換した座標を出力する歪補正座標変換部と、前記変換された座標位置から前記入力画像範囲を算出す

る補正範囲検出部と、を有する。

この構成で、補間座標を生成するとは、歪補正前の撮像画像（入力画像）の位置を求めるために、出力されるべき補正後の補正画像（出力画像）の位置を前もって生成することである。この補間座標に対して歪補正式を適用して、歪補正前の画像位置（座標）を生成（変換）出力させ、変換された補正前の座標位置から歪補正処理すべき入力画像範囲を算出するようにしている。

或いは、好ましくは、前記歪補正範囲算出部は、座標変換により生成された座標に対し、前記出力画像範囲の4辺の各辺に対応する画素の座標の最大値と最小値および前記出力画像範囲の4頂点に対応する座標の少なくともいずれかから前記入力画像範囲を算出する。この構成で、4頂点とは、歪補正後の出力画像における各辺の頂点を指している。

或いは、好ましくは、前記歪補正範囲算出部は、歪補正処理を行う複数入力信号に対して順次範囲算出を繰り返して前記入力画像範囲を算出する。

この構成では、複数のチャンネルを扱う場合に、複数入力信号に対して同時に歪補正処理を行っている間に、複数入力信号に対してそれぞれ次の処理対象であるブロックラインにつき歪補正範囲算出を順次に繰り返して行うことで、1つの歪補正範囲算出部で複数チャンネル分の範囲算出を行わせることが可能である。

好ましくは、前記範囲算出を繰り返し処理で行い、入力画像範囲に対して歪補正後の画像範囲が所定の範囲になるような補正倍率 M を決定する。

この構成により、補正倍率 M を適宜の値に決定することで、陣笠型歪にも対応可能となる。

好ましくは、前記歪補正範囲算出部は、前記歪補正処理部が歪補正実行中に、次に歪補正を行う画像入力範囲を算出する。

この構成では、歪補正実行中に、次に歪補正を行う画像入力範囲を算出するので、1つのブロックラインの歪補正が終わったときには次の歪補正処理を行う画像入力範囲が分かっており、歪補正処理を次のブロックラインに対して順次スムーズに行なっていくことが可能である。

本発明による画像処理方法は、歪補正処理を行う画像処理方法であって、前記歪補正処理を行うに際して、歪補正処理を行う入力画像範囲を算出する。

このように、歪補正処理を行うに際して、歪補正処理を行う入力画像範囲を算出することにより、歪補正後の画像の出力範囲の一部または全部に相当する領域が入力画像内に占める範囲を算出することで、歪補正によって得られる補正画像は画像出力範囲に対して過不足なく出力することが可能となり、画像出力範囲に対して補正画像がはみ出したり不足したりすることのないようにできる。

本発明による画像処理装置は、画像データに対して歪補正処理を行う歪補正手段を有する画像処理装置において、前記歪補正手段は、補正前の画像データ内において、補正画像の各画素が対応する位置を所定の補正式に従って算出する際、撮像面における空間的な位置を記述可能な座標系において算出する。

この構成では、補正前の画像データ内において、補正画像の各画素が対応する位置を算出する際、撮像面に対応する座標系（即ち２次元空間における位置）を基準にして計算を行う。

好ましくは、前記歪補正手段は、補正画像に含まれる第１の小領域を単位に処理を行い、前記第１の小領域の各画素の座標位置を前記補正式に従って変換した小領域を含む撮像面における第２の小領域について、前記撮像面に対応する座標系での位置を、前記第２の小領域での座標に変換した後に、前記補正画像の各画素のデータを生成する。

この構成では、変換される小領域内での座標に基づいて、補間処理を行うことができる。

なお、前記画像データは、撮像データの一部のみを取り込んだ部分画像データであってもよい。

画像データとしては、撮像したデータ全てではなくその一部、例えばデジタルズームの際に撮像データの中央部分から切り出されるデータや、撮像データから間引きによって生成される縮小データに対しても、適用することができる。

また、前記画像データは、前記撮像データに対してフィルタリング、空間サン

ブリングまたは補間処理の少なくともいずれか1つを行って生成されるデータであってもよい。

画像データとしては、周波数的に制限するフィルタリングを行ったり、撮像デバイスの撮像面のような2次元的空间から適宜にサンプリングしたり、補間処理（標本化処理などで離散化した画素と画素の間に、周囲の画素からの推定によって画素を作り出す処理）したり、することによって得られるデータなどがある。

本発明による歪補正方法は、画像データに対して歪補正処理を行う歪補正手段を有する画像処理装置の歪補正方法であって、前記歪補正方法は、補正画像の各画素に対応する座標系での第1の位置を生成するステップと、前記第1の位置を、歪補正式に従って歪補正前の画像データ内における座標系での第2の位置に変換するステップと、前記歪補正前の画像データ内における座標系での第2の位置を、撮像面に対応する座標系を基準とする設定領域内での座標に変換するステップと、前記設定領域内での座標に基づいて補間処理を行い、前記補正画像の各画素のデータを生成するステップと、を有する。

この歪補正方法では、例えば撮像レンズによる幾何学的歪みを補正する補正式によって、補正画像の各画素のデータを得る際には、補正画像の各画素に対応する座標系での第1の位置に対して、歪補正式に従って歪補正前の画像データ内における座標系での第2の位置に変換し、この変換された第2の位置を、撮像面に対応する座標系を基準とする設定領域内での座標に変換し、この座標に基づいて補間処理を行って補正画像の画素データを生成する。

好ましくは、前記補正画像に含まれる小領域を単位に処理を行う。

この方法では、補正画像に含まれる小領域（例えばブロックラインと呼ばれる領域）を単位に処理を行うものである。

本発明による画像処理装置は、画像データに対して歪補正処理を行う歪補正手段を有する画像処理装置において、前記歪補正手段は、前記画像データの一部を格納するメモリ部と、前記メモリ部に対するデータの書き込みおよび読み込みを制御するメモリ制御部と、を有し、前記メモリ部から読み出した画像データに対

する補間演算を行う。

この構成では、内部バッファとしてのメモリ部に画像データの一部を格納して、そのデータを利用して歪補正処理のための補間演算を行うことができる。

好ましくは、前記メモリ制御部は、前記メモリ部への書き込みの際は、列方向に一行に並んだ一定数分の画素からなる画像データ（ユニットライン：UL）を単位として書き込むように制御し、前記メモリ部からの読み出しの際は、歪補正処理後の画像が前記ULを単位として出力されるように、前記メモリ部に格納された画像データに対して読み出し制御する。

この構成では、メモリ部に対してULを単位として書き込み、読み出しを行うので、メモリ部のバッファ容量としては光学的歪量に応じて数UL分、すなわち少なくとも1UL分の容量があればよく、バスの転送量やメモリの容量を大きく増大させることなく、歪補正を実現できる。

さらに、好ましくは、前記メモリ制御部は、前記ULの最初に処理される画素の座標位置に対して、行方向に対して前側および後側に所定幅の領域（それぞれpreULB、postULB）を設け、前記ULの処理中は他の処理によって前記領域が上書きされないよう制御する。

この構成により、処理に必要なバッファとしてのメモリ部の前記preULB、postULBの領域のデータが上書きされないよう確保できる。

好ましくは、前記画像処理装置は、さらにバッファの空き領域を検出するバッファ空き容量監視回路を有し、該バッファ空き容量監視回路によりバッファ内の空き領域が検出されると、該空き領域へのデータの書き込みを可能とする。

この構成により、データ出力中に入力も可能なパイプライン的な動作を可能とする。

好ましくは、前記メモリ部は、データのリード動作とライト動作が同時に可能な複数のメモリからなり、前記メモリ制御部は、さらに、前記メモリ部へのデータ書き込み制御を行う書き込みアドレス生成回路と、前記メモリ部に格納されている画像データから補間演算に必要なデータを同時に読み出すためのアドレスを生成

する読出しアドレス生成回路と、を有し、前記データ書込み制御は、同時に読み出すデータをそれぞれ異なるメモリに書き込むものである。

この構成によれば、バッファ内にある画像データから、補間演算に必要なデータとして任意の座標の周辺の複数の画素（例えば、16画素）情報を同時に読み出すことが可能となり、演算処理を速やかに行える。

本発明による画像処理方法は、画像データに対して歪補正処理を行う画像処理方法において、前記歪補正処理を行う際には、データの書き込みおよび読み込みが制御されるメモリ部に、前記画像データの一部を格納し、該メモリ部から読み出した画像データに対して補間演算を行う。

この方法では、内部バッファとしてのメモリ部に画像データの一部を格納して、そのデータを利用して歪補正処理のための補間演算を行うことができる。

図面の簡単な説明

図1は本発明の第1の実施の形態の画像処理装置の全体構成を示すブロック図である。

図2は歪補正処理部の構成を示すブロック図である。

図3～図5は歪補正処理部における座標変換の概念図であり、図3は撮像画像データを示す図、図4は補正画像を示す図、図5は補間処理を説明する図である。

図6は画像データの読み出し順序を示す図である。

図7は第1データ順序変換部におけるデータ順序変換処理を説明する図である。

図8は第2データ順序変換部におけるデータ順序変換処理を説明する図である。

図9はブロックラインと歪補正処理に必要なメモリ容量（バッファ容量）の関係を示す図である。

図10はブロックライン幅の設定方法を示す図である。

図11は歪補正範囲算出部の構成を示すブロック図である。

図 1 2 及び図 1 3 は歪補正範囲算出部における、歪補正処理を行う際の入力画像範囲を算出する動作を説明する図であり、図 1 2 は補正画像上の範囲を示す図、図 1 3 は元データ上の範囲を示す図である。

図 1 4 は歪補正処理の動作を説明するフローチャートである。

図 1 5 は図 1 4 のステップ S 11 における補正倍率 M の算出方法を説明するフローチャートである。

図 1 6 は図 1 1 の歪補正範囲算出部の構成を示すブロック図である。

図 1 7 は図 1 6 のデータ出力のタイミングチャートである。

図 1 8 は図 1 7 の動作を改善したもので、バイブラインでのデータ出力のタイミングチャートである。

図 1 9 は 3 チャンネル分の歪補正処理を行う際の入力画像範囲を算出する方法を説明する図である。

図 2 0 は 3 チャンネルの場合の歪補正処理の動作を説明するフローチャートである。

図 2 1 は本発明の第 2 の実施の形態の画像処理装置の全体構成を示すブロック図である。

図 2 2 は陣笠型の歪を生じた撮影画像を示す図である。

図 2 3 及び図 2 4 は補正前の撮像画像データと補正後の補正画像の関係を示す図であり、図 2 3 は補正前の撮像画像データを示す図、図 2 4 は補正後の補正画像を示す図である。

図 2 5 は図 2 3 及び図 2 4 の歪補正処理のフローチャートである。

図 2 6 及び図 2 7 は、デジタルズームの場合で、補正前の撮像画像データと補正後の補正画像の関係を示す図であり、図 2 6 は補正前の撮像画像データを示す図、図 2 7 は補正後の補正画像を示す図である。

図 2 8 及び図 2 9 は、撮像データから或る領域を切り出す場合で、補正前の撮像画像データと補正後の補正画像の関係を示す図であり、図 2 8 は補正前の撮像画像データを示す図、図 2 9 は補正後の補正画像を示す図である。

図30～図32はCCDのモニターモードにおけるライン間引きを説明する図であり、図30はCCDから撮像時のデータを縦方向に3ライン間引いてメモリに取り込む状態を説明する図、図31はメモリに取り込まれた状態の撮影画像データを示す図、図32は縦と横で画像が非対称となるのを補正した補正画像を示す図である。

図33及び図34はYC422形式のデータにおける色データの横方向の間引きを説明する図であり、図33は輝度データを示す図、図34は色データを示す図である。

図35は画像の中心ずれを説明する図である。

図36はCCDの画素と撮影画像の関係を示す図である。

図37は本発明の第3の実施の形態の画像処理装置における歪補正処理部の詳細な構成を示すブロック図である。

図38は補間回路における補間演算のイメージ図である。

図39は歪補正処理部における内部メモリ部を説明する図である。

図40及び図41は、図39の補足説明をする、2ポートSRAMにデータが書き込まれる様子を説明する図であり、図40は書き込む順番を示す図、図41は図40に示した書き込み順番のデータが16個の2ポートSRAM上のどこに書き込まれていくかを示す図である。

図42は図38の補正座標位置を算出するのに必要なD0を求めるための説明図である。

図43はエラー処理例を示す図である。

図44は他のエラー処理例を示す図である。

図45は他のエラー処理例を示す図である。

図46は歪補正処理に要するバッファ量について説明する図である。

図47は歪補正処理に要するバッファ量について説明する図である。

図48及び図49は歪補正処理に要するバッファ量について説明する図であり、図48はpreULB、postULBの定義を説明する図、図49は歪中

心を跨ぐ時のバッファ開放量を説明する図である。

図50はUL処理に伴う開放量の算出方法について説明する図である。

図51はUL処理に伴う開放量の算出方法について説明する図である。

図52は図51の処理に伴うパイプライン処理の空きについて説明する図である。

図53はUL処理に伴う開放量の算出方法について説明する図である。

図54は図53の処理に伴うパイプライン処理の空きについて説明する図である。

図55はUL処理に伴う開放量の算出方法について説明する図である。

図56は格子状の被写体を示す図である。

図57は樽型歪を生じた撮影画像を示す図である。

図58は糸巻き型歪を生じた撮影画像を示す図である。

図59は一般的なデジタルカメラの画像処理手順の概念を示す図である。

図60は従来のデジタルカメラの画像処理装置におけるブロック構成を示す図である。

図61は樽型歪を生じた撮影画像を、[式5]によって補正する動作を説明する図である。

図62は糸巻き型歪を生じた撮影画像を、[式5]によって補正する動作を説明する図である。

発明を実施するための最良の形態

発明の実施の形態について図面を参照して説明する。

図1は本発明の第1の実施の形態の画像処理装置の全体構成を示すブロック図である。

図1の画像処理装置においては、バス3に接続された各部の制御を行うCPU4の制御により、CCD1からの撮像信号をプリプロセス回路2で画素欠陥処理、A/D変換等を行い、得られた画像データを、バス3を介してフレームメモリ5

に一旦記憶する。フレームメモリ5は、SDRAM等で構成され、画像処理前のデータ及び画像処理後のデータを格納するメモリである。次いで、フレームメモリ5から読み出した画像データをバス3を介して、第1データ順序変換部6に入力する。第1データ順序変換部6は、後に図7で説明するように、ブロック単位のデータを記憶可能なメモリを複数、ここでは2つ備えている。第1データ順序変換部6は、フレームメモリ5から行方向にデータを読み出して記憶した後、列方向にデータを順に読み出し、イメージプロセス回路7に出力する。

イメージプロセス回路7は入力したデータに対して所定の画像処理を行って次段の歪補正手段としての歪補正処理部8に転送する。歪補正処理部8は入力したデータに対して歪補正処理を行って次段の第2データ順序変換部9へ転送する。第2データ順序変換部9は、後に図8で説明するように、ブロック単位のデータを記憶可能なメモリを複数、ここでは2つ備えている。第2データ順序変換部9は、歪補正処理部8から列方向にデータを読み出して記憶した後、行方向にデータを順に読み出し、JPEG処理部10に転送する。そして、JPEG処理部10でJPEG圧縮処理を行い、処理データをフレームメモリ5に一旦記憶し、フレームメモリ5から読み出した処理データをメモリカード等11に記録するようになっている。

歪補正処理部8は、図2に示すように、歪補正後の補正画像の位置（補間位置という、 X 、 Y ）とこれに対応した歪補正前の元の画像の位置（ X' 、 Y' ）を生成する補間座標生成部81と、歪補正処理部8の前段ブロックの回路からの画像データの一部を一時的に格納しておくバッファメモリ（以下、単にバッファ）としての内部メモリ部82と、その内部メモリ部82に対する書き込み及び読み出しの制御を行うメモリ制御部83と、変換した歪補正前の画像位置の座標（ X' 、 Y' ）に従って画像処理をしてデータの歪補正をする補間演算部84と、を有して構成されている。

補間座標生成部81は、図2に示すように、補間座標（ X 、 Y ）を生成する補間位置生成部811と、生成された補間座標（ X 、 Y ）に対して所定の歪補正式[式

1] (後述する) を適用して変換した補正前の座標 (X' , Y') を出力する歪補正座標変換部 812 と、補間位置生成部 811 からの補間座標 (X , Y) と歪補正座標変換部 812 からの変換座標 (X' , Y') とを選択的に出力可能なセクタ 813 と、で構成されている。歪補正処理部 8 における補間位置生成部 811, 歪補正座標変換部 812, セクタ 813, メモリ制御部 83 については、制御データを格納した制御レジスタ 85 に設定された各ブロックに対する設定値に従って動作する。また、処理結果のステータスなどを CPU から参照することができる。

歪補正処理部 8 には、図 1 に示すように、歪補正処理部 8 が歪補正処理を行う入力画像範囲を算出する歪補正範囲算出部 12 が設けられている。歪補正範囲算出部 12 は、図 11 で説明するように、補間座標を生成する座標生成部 91 と、生成された補間座標に対して所定の歪補正式を適用して変換した座標を出力する歪補正座標変換部 92 と、変換された補正前の座標位置から前記入力画像範囲を算出する補正範囲検出部 93 と、を有して構成されている。

以上のように構成された本発明による画像処理装置においては、上記第 1 データ順序変換部 6 から JPEG 処理部 10 までは、バス 3 を介することなく、該バス 3 とは異なる情報伝達経路でパイプライン処理可能なように接続されていて、画像データを、2 次元的な画素配列における所定のブロック単位で転送して処理するようになっている。このようにバス 3 を介したデータ転送は、フレームメモリ 5 から第 1 データ順序変換部 6 への転送と、JPEG 処理部 10 からフレームメモリ 5 への転送及びフレームメモリ 5 からメモリカード 11 への転送だけとなり、したがってフレームメモリと各イメージプロセス回路との間でデータのやりとりを行っていた従来例 (図 60) と比べて、バス 3 によるデータ転送量を大幅に低減させることができ、バス 3 の負荷を大幅に軽減することが可能となる。また、この図 1 に示す例においては、画像処理を行うイメージプロセス回路 7 を 1 つのみ設けているが、複数あっても構わない。さらに、図では、イメージプロセス回路 7 の後に歪補正処理部 8 が設けられているが、逆の構成であっても構わない。

初段のイメージプロセス回路 7 と 2 段目のイメージプロセス回路である歪補正処理部 8 で構成されるイメージプロセス回路部では、各イメージプロセス回路 7, 8 の前段或いは内部にパイプラインレジスタとして、図示しない小容量のメモリが配置されていて、該小メモリを介して、各イメージプロセス回路 7, 8 がパイプライン処理動作を行うように構成されている。これらの小容量のメモリは、各イメージプロセス回路 7, 8 で空間的な画像処理を行う場合、画像処理に必要な周辺データの記憶をするために、また画像データをブロック単位で読み出して配列換えなどを行って処理する必要があるために、設けられている。

図 3 ～ 5 は歪補正処理部 8 における座標変換の概念図を示している。図 3 は元データである撮像画像データ、図 4 は補正画像、図 5 は図 4 の補正画像の座標位置 (X, Y) に対して図 3 の元データの座標上に変換された座標位置 P でのデータの座標 (X', Y') (この座標 P は元データを実際に構成する複数の画素の位置には必ずしも正確には一致しない座標位置にある) を示している。その P 点での座標位置 (X', Y') はその P 点の周辺の画素 16 点の座標を使って算出されると共に P 点での画像データはその周囲の前記 16 点の画素データを用いて補間演算される。補間演算部 84 にて点 P の位置のデータをその周囲の 16 点の画素値 (輝度データ) から補間演算するべく処理が行われる。図 2 の補間位置生成部 811 で補間座標を生成するというのは、図 4 の補正画像側でどこの画素位置 (X, Y) を指し示すか、ということである。

[式 1] では、歪補正後の画素位置 (X, Y) に対する歪補正前の画素位置 (X', Y') を算出することができる。但し、歪補正前の画素位置 (X', Y') は元画像データ上の画素位置に対応した整数値になるとは限らないことは前述した通りである。

[式 1] に従うと、図 4 の座標 (X, Y) が図 3 のように座標 (X', Y') に変換される。これにより、元データの中のどこの座標位置のデータを作成すれば、よいかを知ることができ、その位置 (X', Y') のデータを 16 点補間処理を行う補間式を用いて周囲 16 点の既知の画素値 (図 5 の黒丸印の入力画像データ)

から算出して得ることができる。

[式1]

$$\begin{cases} \dot{X} = M \cdot (X - X_d) + (X_d + X_{off}) \\ \dot{Y} = M \cdot (Y - Y_d) + (Y_d + Y_{off}) \end{cases} \dots (1)$$

$$Z^2 = [S_x \cdot (\dot{X} - X_d)]^2 + [S_y \cdot (\dot{Y} - Y_d)]^2 \dots (2)$$

$$\begin{cases} X' = (\dot{X} - X_d) \cdot [1 + A \cdot Z^2 + B \cdot Z^4 + C \cdot Z^6 + \dots] + X_d \\ Y' = (\dot{Y} - Y_d) \cdot [1 + A \cdot Z^2 + B \cdot Z^4 + C \cdot Z^6 + \dots] + Y_d \end{cases} \dots (3)$$

(X, Y)	: 補正画像内での座標
(X', Y')	: 撮影画像内での座標
(X_d, Y_d)	: 歪中心位置
(X_{off}, Y_{off})	: 中心ずれ補正
M	: 範囲補正倍率
(S_x, S_y)	: サンプルング比
A, B, C, ...	: 歪補正係数

[式1]におけるZは、歪中心 (X_d, Y_d) から今注目している点 (X, Y) までの距離である。[式1]によって、補正画像の点 (X, Y) に対して歪んでいる元画像の方の座標 (X', Y') が算出される。[式1]について補足説明すると、Mは光学系のデータを用いて理論的に補正した際、補正後に画像がはみ出したり、不足したりする現象を補正するための補正倍率である。 S_x, S_y は間引き取込みなど、縦横で空間的なサンプルング間隔が異なる現象の補正をするためのサンプルング比である。 X_{off}, Y_{off} は歪補正処理によって、歪補正処理後に、被写体位置が撮影時の位置とはずれてしまう現象の補正をする中心ずれの値である。

本発明に係る[式1]では、高次項（具体的には Z^4, Z^6, \dots ）を考慮することで、より複雑な陣笠型の歪（図22参照）にも対応できるようにしている。ま

た、中心が一致していても撮像素子の一部しかデータを取り込まない場合や、光軸中心と撮像素子中心がずれている場合など、歪中心 (X_d , Y_d) を考慮する必要がある。さらに、歪補正処理により被写体位置が撮影時の位置からずれる現象を補正するため、中心ずれ補正值 (X_{off} , Y_{off}) を考慮する必要がある。間引き取込みの場合や輝度データ (Y) と色データ (C_b , C_r) からなるYC画像を処理する場合は、 X , Y で空間的なサンプリングが異なる。そこで係数としてサンプリング比 (S_x , S_y) を導入して、座標変換を補正するようにしている。

図61の樽型歪では歪補正するために $M < 1$ として少し縮むようにしておかなければならない。図62の糸巻き型歪の場合は逆に引き伸ばさなければならないので、 $M > 1$ に設定する。

従来例で述べた [式5] では、樽型歪・糸巻き型歪を補正することを考慮して記述されている。

次に、図6～図8を参照して画像データの書き込み及び読み出しの処理及び順序について説明する。

図6は、本実施の形態におけるフレームメモリからの画像データの読み出し順序を説明する図である。

画像データは、通常は、ライン方向、つまり行方向にスイープさせて書き込まれていて、読み出し時は行方向に読み出されるようになっており、1ラインの画像データを全て読み出して、次に隣接するラインの画像データを全て読み出すといった動作を繰り返して行うのが普通である。

これに対して、本発明に係る画像処理装置は、行方向にスイープさせて書き込まれた画像データを、列方向にある一定の長さを単位に、行方向の画像データを順番にイメージプロセス部7に入力し、以降、順次隣の列をイメージプロセス部7に入力して、画像の右端まで繰り返し入力するようにしており、こうして得られる小領域（矩形状の画像データ）をブロックライン (BL) と呼ぶ。

こうした画像データの読み出しを可能にするための第1データ順序変換部6

と、第2データ順序変換部9との構成を、図7及び図8を参照して説明する。図7及び図8は、第1、第2データ順序変換部の構成を示すブロック図である。

第1データ順序変換部6は、図7に示すように、ブロック単位の画像データを記憶可能なメモリを複数、ここでは2つ備えており、この2つのメモリ6aとメモリ6bが書き込み側、読み出し側の各スイッチにて書き込み、読み出しが交互に切り換えられるようになっている。すなわち、フレームメモリ5は、書き込み側のスイッチにて、これらメモリ6aとメモリ6bとに切換可能に接続されているとともに、イメージプロセス部7も読み出し側のスイッチにてこれらのメモリ6aとメモリ6bとに切換可能に接続されていて、フレームメモリ5がメモリ6aとメモリ6bとの一方に接続されているときには、該メモリ6aとメモリ6bとの他方がイメージプロセス部7に接続されるように切り換えられる。すなわち、メモリ6a、6bは、フレームメモリ5とイメージプロセス部7との両方に同時に接続されることがないようにスイッチングされ、交互に書き込み、読み出しが行われるようになっている。

フレームメモリ5に記憶されているフレーム画像の一部は、ブロック単位でライン方向に読み出され、一方のメモリ、ここでは例えばメモリ6aに記憶される。

これと並行して、メモリ6bからは、既にフレームメモリ5から読み出されて記憶されているブロック単位の画像データが、列方向（縦方向）に順に読み出されて、イメージプロセス部7へ出力される。

フレームメモリ5からメモリ6aへの書き込みと、メモリ6bからイメージプロセス部7への読み出しが終了すると、書き込み側のスイッチおよび読み出し側のスイッチが切り換えられて、次に、フレームメモリ5からメモリ6bへ次のブロック単位の画像データの書き込みが開始されるとともに、メモリ6aからイメージプロセス部7へのブロック単位の画像データの読み出しが開始される。

第2データ順序変換部9も、図8に示すように、上記第1データ順序変換部6とほぼ同様に構成されていて、ほぼ同様に動作するようになっている。

すなわち、第2データ順序変換部9は、メモリ9a、メモリ9bと、書き込み

側スイッチ、読み出し側スイッチとを有して構成されている。

そして、該第2データ順序変換部9の動作時には、歪補正処理部8からの書き込みがメモリ9aとメモリ9bとの一方に対して列方向（縦方向）に行われ、メモリ9aとメモリ9bとの他方からは行方向（横方向）に読み出しが行われて、JPEG処理部10へ出力されるようになっている。

図9はブロックラインと歪補正処理に必要なメモリ容量（バッファ容量）の関係を示している。点線枠上の歪曲した4本の実線は、歪補正後の出力データでは直線（縦1ライン）となるべきデータであるが元データでは歪んでいる。4本の直線のうち、画像中心から離れた左端の直線がもっとも歪曲している。黒丸印は歪補正後の出力データの画素位置に対応する歪補正前の位置を示している。前述の内部メモリ部82では、撮像後の歪んだ入力データがブロックライン内で最大の歪量となる横方向の幅に16点補間を考慮して左右に余裕分をとった範囲を、歪補正処理に必要なバッファ量として確保するようにする。すなわち、これだけのバッファ容量があれば、歪補正可能であるバッファ量、換言すれば歪補正したときに正しい直線形状を作成することが可能なバッファ量を示している。

図10は、ブロックライン幅の設定方法を示すものである。ブロックライン幅を歪補正処理の対象位置に応じて可変設定する例を示している。湾曲した点線は入力側の歪んだデータを示しており、画像中心より離れるほど、すなわち外側にいくほど歪み度合いが大きくなっている。従って、フレームメモリ上の入力データに対してブロックライン幅を設定する際には、その幅は画像中心から離れるほど大きく、中心に近いほど幅を狭くするように設定する。これにより、フレームメモリ5からバス3を介してイメージプロセス回路にデータ転送する際に、データ幅が狭くてよければそのときのバスの専有時間を減らすことができる。

一般的に、画像中心では歪による変形量が小さい。歪補正の処理対象の位置によって、ブロックライン幅（図示の縦方向にスweepする幅）の設定値を変更する。式1には高次項が含まれるため、ブロックライン分の処理に必要な入力範囲を解析的に求めることができない。ブロックライン幅の設定は、歪補正範囲算出

部 1 2 の処理結果に基づき、また、CPU で該結果に所定の演算を施すことにより設定する。

図 1 1 は、歪補正範囲算出部 1 2 の構成を示している。歪補正範囲算出部 1 2 は、補間座標 (X, Y) を生成する座標生成部 9 1 と、生成された補間座標 (X, Y) に対して所定の歪補正式 (例えば[式 1]) を適用して変換した座標 (X', Y') を出力する歪補正座標変換部 9 2 と、変換された補正前の座標位置 (X', Y') から歪補正処理に必要な入力画像範囲を算出する補正範囲検出部 9 3 と、を有して構成されている。このように歪補正範囲算出部 1 2 は、座標生成部 9 1、歪補正座標変換部 9 2、補正範囲検出部 9 3 からなり、CPU 4 からは制御レジスタ 9 4 を通じて動作を制御し、範囲算出結果はレジスタ 9 5 を通じて取得する。歪補正範囲算出部 1 2 は、歪補正処理機能に対して付加したもので、歪の変形を考慮して画像データの入力範囲を算出可能とするサポート機能として働く。

次に図 1 2 及び図 1 3 を参照して、歪補正範囲算出部 1 2 における、歪補正処理に必要な入力画像範囲を算出する動作を説明する。

補正画像内で範囲を指定し、入力画像上に座標変換する方法で行う。まず、図 1 2 に示すように補正画像上の所定範囲 (例えば 5 行分) の画素位置 (格子点位置) を決め、[式 1] にて図 1 3 に示すように出力側の元データ上に座標変換することで、出力側データでの 5 行分のデータの範囲が分かる。処理の設定は、ハードウェアのレジスタ設定で行う。図 1 2 の補正画像上で、例えばスタートは、点 (X_{st}, Y_{st}) から間隔 ΔW , ΔH という順番で格子点を生成していき、縦に何点、横に何点かとり、縦に H_{out} 画素、横に W_{out} 画素となるように設定する。そして、元データ上の位置に変換した結果、補正画像の格子点に対応した交点座標が歪曲した線上に形成されるが、そのブロックラインに対応した歪曲した範囲すなわち外接する図示の太い点線が入力画像範囲として分かればよい。実際には、図 1 2 及び図 1 3 で黒丸印にしてあるような矩形状の周囲 4 辺上の黒丸位置の座標だけ追っていけば、その位置範囲を規定する 4 辺について最大値と最小値から入力画像範囲が分かる。例えば、上の辺は Y_{Tmax}, Y_{Tmin} という間に、下の

辺はY Bmax, Y Bmin という間に収まり、左の辺はX Lmax, X Lmin という間に、右の辺はX Rmax, X Rmin という間に収まっている、というデータを実際に検出して、結果格納用レジスタ95に格納する。ブロックライン処理には、(X Lmin ~ X Rmax, Y Tmin ~ Y Bmax) に補間に必要な画素を加えた範囲が必要である。なお、図13で、出力側の元データ上で、各頂点の変換後の位置を(X' TL, Y' TL)、(X' TR, Y' TR)、(X' BL, Y' BL)、(X' BR, Y' BR)としている。

座標生成部91は、[式2]で補正画像に必要な座標位置(X, Y)を生成し、[式1]で(X', Y')に変換する。上辺の変形範囲をY Tmin ~ Y Tmax、下辺の変形範囲をY Bmin ~ Y Bmax、左辺の変形範囲をX Lmin ~ X Lmax、右辺の変形範囲をX Rmin ~ X Rmax、各頂点の変換後の位置を(X' TL, Y' TL)、(X' TR, Y' TR)、(X' BL, Y' BL)、(X' BR, Y' BR)とし、これらを補正範囲検出部93で検出し、結果格納用レジスタ95に格納する。

[式2]

$$\begin{cases} X = X_{st} + k \cdot \Delta W \\ Y = Y_{st} + l \cdot \Delta H \end{cases}$$

k, lは整数

図14は、歪補正処理の動作を説明するフローチャートである。まず、ステップS11で、歪補正[式1]の補正倍率Mを決定する。補正倍率Mを決定の仕方は図15のフローチャートにて後述する。次に、ステップS12では、決定した補正倍率Mを使って、図10~図13の説明で述べたようにブロックライン処理に必要な入力範囲を補間に必要な部分も考慮して算出し、ステップS13でこの算出した入力範囲と、歪補正処理に必要な設定値を設定する。そして、ステップS14で、ブロックライン毎の歪補正処理と次ブロックラインの入力範囲算出を同時実行する。その後、画像の最下段のブロックラインの歪補正処理が行われるまで、ブロックライン毎の歪補正処理と次ブロックラインの入力範囲算出を実行す

る。ステップS 15 において出力画像全体について歪補正処理が終了したか否かを判定することで、ステップS 14 を繰り返し、画像全体について歪補正処理が終わるまで続行して終了する。

このようにすれば、歪補正実行中に、次に歪補正を行う画像入力範囲を算出するので、1つのブロックラインの歪補正が終わったときには次の歪補正処理を行う画像入力範囲が分かっており、歪補正処理を次のブロックラインに対して遅れなく順次スムーズに行なっていくことが可能である。

上記のステップS 11 で、補正倍率Mを適切に決めてやると、図6 1及び図6 2で述べた樽型及び糸巻き型のどの歪み画像を補正処理しても画像の出力範囲にきっちりと入れることが可能である。

図1 5は図1 4のステップS 11 における補正倍率Mの算出方法を説明するフローチャートを示している。まず、ステップS 21 で補正倍率Mの初期値として1.0を設定する。次にそのMと歪補正[式1]を用いて出力画像の例えば4辺を座標変換することで歪補正範囲算出部1 2にて入力画像範囲を算出し（ステップS 22）、元画像の範囲内か否かを判定する（ステップS 23）。ステップS 23 で図6 1のように出力範囲を越えるようであれば、Mを ΔM だけ減少させた後（ステップS 24）、ステップS 22 に戻り、入力画像範囲を算出し、再びステップS 23 の範囲内判定を行う。すなわち、少しずつMの値を変えて、ステップS 23 で元画像の範囲内か否かを判定し、元画像の範囲内であれば、ステップS 25 へ移行する。ステップS 25 では前記の入力画像範囲が元画像の範囲内に収まる最大のMか否かを判断する。元画像の範囲内で最大でないならば、Mを ΔM だけ増加させた後（ステップS 26）、ステップS 22 に戻り、ステップS 23 ～ 26 を行う。ステップS 25 で元画像の範囲内で最大ならば、そのときのMを補正倍率として決定する。このとき、範囲算出に用いられる領域は、補正画像の全体である。

なお、図1 5のフローは、他の方法でも良い。例えば、補正倍率Mの最大及び最小 M_{\max} 、 M_{\min} を設定し、順次その範囲を2分割して最適なMを追い込む方法を用いると、計算量が少なくて済む。

ところで、図 1 4 におけるステップ S 14 において、歪補正処理部 8 の補正演算に比べて、歪補正範囲算出部 1 2 の範囲算出演算の方が演算量が少ない。そこで、補正演算の乗算器を時系列で使うことで、回路規模の削減が可能となる。

歪補正処理を行う際には、[式 1]を見れば分かるように非常に掛算が多くてハードウェア的には回路規模が非常に大きい計算になっている。一般的に乗算器がハードウェアの回路規模を大きくしているので、出来るだけ乗算器を減らすことが好ましい。乗算器制御部を設けて乗算器制御し、タイミングを取りながら、パイプライン的に処理することで、全体の処理速度を下げずに、回路規模を縮小できる。

図 1 6 は、図 1 1 の歪補正座標変換部 9 2 の部分の具体的な構成を示している。それ以外の部分は、図 1 1 と同様である。歪補正座標変換部 9 2 では、[式 1]のような乗算の多い計算を行うが、その際の複数の乗算をタイミングを図りながら時系列で処理することで、乗算器数を減らすことができるように構成してある。

歪補正座標変換部 9 2 は、座標生成部 9 1 から補間座標 (X , Y) を入力し、[式 1]の (2) を演算して Z^2 を算出する Z^2 算出部 921 と、 Z^2 算出部 921 からの Z^2 と乗算器 923 からの γ を入力し、 α , β を出力する乗算器制御部 922 と、 α , β を入力し、 γ を出力する乗算器 923 と、補正係数 A , B , C , ... のいずれかに相当する δ を出力する乗算器制御部 924 と、 α , δ を入力し、補正係数と、 Z の整数倍との積に相当する ε を出力する乗算器 925 と、乗算器 925 からの ε と座標生成部 9 1 からの補間座標 (X , Y) を入力し、補正前の元データの座標 (X' , Y') を出力する補正座標算出部 926 と、を有して構成されている。

図 1 7 は図 1 6 のデータ出力のタイミングチャートを示している。まず、 Z^2 算出部 921 は、クロック CLK が '1' のタイミングで補間座標 (X , Y) を入力する。次に乗算器制御部 922 は、クロック CLK の '2' のタイミングで α , β として Z^2 を出力し、乗算器制御部 924 は同じタイミング '2' で δ として歪補正係数 A を出力する。次のクロック CLK の '3' のタイミングで、乗算器 923 は入力される α , β に基づいて γ として Z^4 を出力し、乗算器 925 は入力

される α 、 δ に基づき同じタイミング‘3’で ε として AZ^2 を出力する。同様に、続くクロックCLKの‘4’、‘5’のタイミングでは、 δ 、 ε として B 、 BZ^4 を得、クロックCLKの‘6’、‘7’のタイミングでは、 δ 、 ε として C 、 CZ^6 を得る。その結果として、クロックCLKの‘8’のタイミングで、補正座標算出部 926 から補間座標 (X, Y) に対する元データの座標位置 (X', Y') を出力する。

図16の構成では、乗算器は2つしかないが、乗算器を時系列で使うことで、実際には6つ分の計算をさせている。結果的に乗算器を $1/3$ に減らすことが可能となる。さらに高次項を考慮する場合でも、同様の動作をさせればよい。

ところで、図17のタイミングチャートを見ると、歪補正座標変換処理においてクロックごとの各変数の出力データには時間的な隙間（空き）があるのが分かる。そこで、パイプライン処理により各変数の出力データを時間的に空きのないように連続的に出力させることが考えられる。

図18は、図17の動作を改善したものであり、パイプライン処理でのデータ出力のタイミングチャートを示している。図17の動作タイミングでは、補間座標 (X, Y) の入力から7クロック経過後に出力として元データの座標位置 (X', Y') が得られている、つまり、1つ結果が出るまでに7クロック分の時間を要する。これに対して、図18では、10サイクル目に1サイクル目と同じことが起こるので、9サイクルで3画素処理していることになる。つまり、見かけ上3サイクルで1画素処理していることになる。

これまでは、歪補正範囲算出部では1つのチャンネルについてしか歪補正範囲算出処理を行わない例について述べてきたが、次に前出の1チャンネル分にかかった時間と同じ時間で、3チャンネル分を1つの歪補正範囲算出部で行う方法について説明する。これには、図19に示すように、図12の補正画像上の範囲設定において、注目領域の4辺における範囲算出用画素を例えば $1/3$ の画素数に間引いて設定し演算処理させることで、1チャンネル分の処理時間とほぼ同じ時間で終わることができる。間引き量はレジスタ設定で行う。図19のように間引

いた座標（黒丸を付したところ）でしか演算しないため、前出の処理時間と同じ時間で複数のチャンネル分に対応でき、しかも回路規模を増大させることがない。そうすると、例えばカラー画像を扱う場合に、R，G，Bごとに歪補正範囲算出部を設けなくとも3チャンネル分を1つの回路部で動作させれば1つの歪補正範囲算出部で済み、回路規模を増大させることなく、同じ時間内で済ませることができる。

図20は、3チャンネルの場合の歪補正処理の動作を説明するフローチャートである。図14に示した歪補正処理を3チャンネル分に適用したものである。

図20におけるステップS31～S33及びステップS35は、図14のステップS11～S13及びステップS15と同じである。

ステップS34では、3チャンネルにおけるブロックライン毎の歪補正処理と、1チャンネル目（例えば、R）に対する次ブロックラインの入力範囲算出→2チャンネル目（例えば、G）に対する次ブロックラインの入力範囲算出→3チャンネル目（例えば、B）に対する次ブロックラインの入力範囲算出に対する入力範囲算出とを同時に実行する。

これにより、複数のチャンネルを扱う場合に、複数入力信号に対して同時に歪補正処理を行っている間に、複数入力信号に対してそれぞれ次の処理対象であるブロックラインにつき歪補正範囲算出を順次に繰り返して行うことで、1つの歪補正範囲算出部で複数チャンネル分の範囲算出を行わせることが可能である。また、歪補正実行中に、次に歪補正を行う画像入力範囲を算出するので、1つのブロックラインの歪補正が終わったときには次の歪補正処理を行う画像入力範囲が分かっており、歪補正処理を次のブロックラインに対して遅れなく順次スムーズに行なっていくことが可能である。

以上述べたように本発明の第1の実施の形態によれば、歪補正処理を行う入力画像範囲を算出する歪補正範囲算出部を設けたので、歪補正によって得られる補正画像は出力すべき出力範囲に対して過不足なく出力することが可能となる。元データを有効に活用して歪補正処理を行え、糸巻き型歪、樽型歪、及び陣笠型歪

にも対応できる歪補正処理が可能となる。また、ブロックライン毎の処理に必要な入力範囲を算出することができ、データ転送量を最小限にすることが可能となる。

図 2 1 は本発明の第 2 の実施の形態の画像処理装置の全体構成を示すブロック図である。図 2 1 は図 1 から歪補正範囲算出部 1 2 を削除した構成となっている。図 2 1 における歪補正処理部の構成は図 2 と同様である。

歪補正座標変換の [式 1] は、あくまでも全画面の中の画像であって、補正画像についてもその座標変換結果の (X' , Y') の画像にしても、全画面中で何処に位置しているかを計算している。

ところが、歪補正処理部 8 に入力される画像データはブロックライン単位である。歪補正処理部 8 としては、入力される画像データの空間的な位置関係 (2 次元空間における位置) を知る必要がある。制御レジスタ 8 5 は、歪補正処理部 8 で処理するブロックラインの位置を指定している。

図 2 3 及び図 2 4 は、補正前の撮像画像データと補正後の補正画像の関係を示している。図 2 3 は撮影画像データ、図 2 4 は補正画像を示している。図 2 3 及び図 2 4 は補正画像 (出力側画像) の或るブロックライン BL の或る画素位置 A に注目している場合の概念図である。補正画像 (出力画像) 内の位置 A の座標 (X , Y) は、歪補正座標変換の [式 1] を用いて (入力画像) 内の位置 A' の座標 (X' , Y') に変換される。

座標位置 (X , Y) 及び (X_{blst} , Y_{blst})、座標位置 (X' , Y') 及び (X'_{blst} , Y'_{blst}) は、全画面の原点からの位置を示している。ここでは、原点は、全画面の左上端部としている。

歪補正ブロックへ入力される画像データは、符号 BL にて示すブロックライン単位である。そのため、まず補正画像の例えば画素 A を歪座標変換処理するには、最終的な補間処理ができるようにブロックライン BL 内での座標系における座標位置 (x' , y') に変換する必要がある。

まず、図 2 4 の補正画像 (出力画像) の各画素に対する座標位置 (X , Y) は、

下記の〔式 3〕で生成する。

〔式 3〕

$$X = X_{\text{blst}} + m \times \Delta X$$

$$Y = Y_{\text{blst}} + n \times \Delta Y$$

ΔX , ΔY は補間ピッチで、1.0以上なら縮小画像が生成され、1.0以下なら拡大画像が生成される。座標位置 (X_{blst} , Y_{blst}) は歪補正ブロックの出力側ブロックラインの左肩の位置、すなわち、ブロックラインのスタート位置に相当している。 $(m, n) = (0, 0)$ のとき、 $X = X_{\text{blst}}$, $Y = Y_{\text{blst}}$ となるので、〔式 3〕から座標位置 (X_{blst} , Y_{blst}) を算出できることになる。 $m = 0$, $n = 0$ から始めて、 m , n を例えば交互に 1 ずつ増やしていけば、ブロックライン BL 内の全ての画素の座標位置 (X , Y) を生成することができる。

この BL に対し、BL 内全ての画像の座標位置を〔式 1〕に従って変換すると、図 23 の変形したハッチング領域 BL' にマッピングされる。図 23 の BL' は、図 24 の BL を処理するのに必要な入力データの範囲となる。 $(X'_{\text{blst}}$, Y'_{blst}) は、BL' の左上位置であり、補間に必要な画素等を考慮すると、変形したハッチング領域の左上位置とは一致しない。BL' は、図 1 で説明したサポート機能 (歪補正範囲算出部 12) を利用して設定する。

〔式 3〕で生成された補正後の出力画像の座標位置 (X , Y) を、〔式 1〕で座標変換して図 23 の補正前の入力画像の座標位置 (X' , Y') を求める。そして、座標変換された座標位置 (X' , Y') と、座標位置 (X'_{blst} , Y'_{blst}) との差分をとることにより、ブロックライン (太い点線の矩形) の左肩を原点とする座標位置 (x' , y') を求めることができる。

その結果、このブロックライン内での座標位置 (x' , y') に基づき、ブロックライン単位での補間処理を行えることになる。

図 23 に示す入力画像側の太い点線で示すブロックライン BL' はレジスタ 85 によって設定するが、その値を生成するためには、歪の変形を考慮して画像データの入力範囲 (歪補正範囲) を算出可能とするサポート機能が必要となる。こ

のサポート機能は、歪補正処理部 8 の歪補正処理機能に付加するなどして設けられよう。

次に、図 2 5 を参照して歪補正処理フローを説明する。図 2 5 は図 2 3 及び図 2 4 の歪補正処理のフローチャートである。

まず、図 2 4 のハッチングを施したところのブロックライン処理を開始する(ステップ S 41)。前述の[式 3]で、 $(m, n) = (0, 0)$ とおくことにより(ステップ S 42)、図 2 4 のハッチング領域の左肩、すなわちブロックラインのスタート座標位置 (X_{blst}, Y_{blst}) を求める。そして、[式 3]から補正画像の座標位置 (X, Y) を求めることができる。この場合、 m, n をそれぞれ 1 ずつ増やしていけば、図 2 4 のハッチングで示すブロックライン BL 内の各画素の座標位置 (X, Y) を生成することができる(ステップ S 43)。

次に、[式 1] を用いて補正画像における座標系の位置 (X, Y) を、補正前の撮影画像における座標系の位置 (X', Y') に変換する(ステップ S 44)。

そして、撮影画像における座標系の位置 (X', Y') を、前述した方法で入力ブロックライン BL 内の座標系の位置 (x', y') に変換する(ステップ S 45)。

次に、このブロックライン BL 内での座標位置 (x', y') に基づいて補間処理を行い、座標位置 (X', Y') における 16 点補間などで周辺の複数の画素データから補間データを算出することで、1 画素分の補正画像データが得られる(ステップ S 46)。

そして、以上のステップが、補正画像の座標系の BL 内全画素について行われたか否かについて判断し(ステップ S 47)、終了していなければ次の m, n を設定して(ステップ S 48)、ステップ S 43 ~ S 47 を繰り返し行い、ステップ S 47 の判定で全画素について処理が終了していれば、ステップ S 49 に移行する。

ステップ S 49 では、全画面(フレーム)内の全てのブロックライン BL について歪補正処理が終了したか否かを判断しており、終了していなければ次の入力ブロックライン BL を選択し、ステップ S 41 ~ ステップ S 49 を繰り返し、ステップ S 49 の判定で全画面について処理が終了していれば、終了する。

図26～図29は、図25の変形例とも言うべきものである。

デジタルズームを行う場合や注目する小領域のみを切り出す場合、撮像データの一部しか使わなくても構わない。選択的にCCDから撮像データの一部を取り込むことでメモリを節約することができる。この場合に歪補正をブロックライン単位で行うためには、取り込まれたデータが、撮像素子上でどこに位置するかの情報 ($X' \text{ imgst}$, $Y' \text{ imgst}$) が必要である。

図26及び図27は、デジタルズームの場合を示し、撮像した画像データから中央部分のみを取り込むものである。図25の場合と異なる点は、上述したように、全画面の中で何処を取り込むかをプリプロセス回路2で設定している点である。この設定に基づいて必要な部分のみを取り込んでおり、その範囲を図26の E' で示す。

まず、図27で全画面の中のデジタルズームする範囲 E の左肩に当たるスタート位置 ($X \text{ imgst}$, $Y \text{ imgst}$) を設定する。次に、ブロックライン BL を設定するが、ここでは梨地にて示す部分である。このとき、取り込み範囲 E の左肩を原点とする、位置座標 ($X \text{ blst}$, $Y \text{ blst}$) が決定し、[式2]と同様に m , n を変化させることで、ブロックライン BL 内における各画素の座標を生成することができる。

この結果、座標位置 ($X \text{ imgst}$, $Y \text{ imgst}$)、($X \text{ blst}$, $Y \text{ blst}$)、($m \times \Delta X$, $n \times \Delta Y$) から全画面の座標における位置 (X , Y) が決定される。

撮影画像における座標位置 (X' , Y') は、上記のように決定された座標位置 (X , Y) から [式1] を用いて求めることができる。

そして次に、撮影画像データにおけるブロックライン BL' の中での座標位置 (x' , y') を求める。

まず、取り込まれた撮像データの撮像面における設定範囲 E' から左肩の座標 ($X' \text{ imgst}$, $Y' \text{ imgst}$) が分かる。取り込み範囲 E' の左肩を原点とする座標で、ブロックライン BL' の左肩の座標位置 ($X' \text{ blst}$, $Y' \text{ blst}$) はサポート機能を利用して予め求められており、ブロックライン BL' 内での座標における座

標位置 (x' , y') が決定される。その結果、ブロックラインごとの補間処理が行われる。

図 2 8 及び図 2 9 は撮像データから或る領域、例えば画面の端の方を切り出す場合である。これは、監視カメラなどにおいて、撮像データの端部分を取り込むような場合に相当する。図 2 8 及び図 2 9 の場合も、図 2 6 及び図 2 7 と同様な符号を付してある。図 2 6 及び図 2 7 と切り出す位置が異なっているだけで、図 2 6 及び図 2 7 の場合と同様な動作となる。

なお、通常のデジタルカメラでは、CPU が制御レジスタに切り出す領域を設定するため、切り出し位置が分かっており、上記の変換が可能だが、記録後のデータに歪補正処理を行う場合ではタグ等に切り出し領域の位置とサイズを同時に記録する必要がある。このとき、同時に補正係数も記録しておく。

ところで、撮像素子としての CCD にはモニターモード（撮像した全画面を読み出さないモード）が用意されることが多い。モニターモードでは、ラインが間引かれることが多く、撮像素子における 2 次元空間的には同一でも、データ上ではサイズが異なることがある。また、プリプロセスにおいて、横方向の画素も間引かれることがある。また、CCD からの撮像データにローパスフィルタ (LPF) などのフィルタリング処理をかけたり、補間処理をしてフレームメモリに取り込むことがある。この場合、縦横のサンプリング比が整数にならない場合が生じる。さらに、JPEG における YCbCr の記録では、Cb や Cr 成分の横方向は 2 倍の間隔でサンプリングされることが多く、データとしては横方向が縮んだ形になる。これらは、[式 1] では、 S_x , S_y によって補正する。

図 3 0 は、撮像時のデータ取り込みにおいて、実際に CCD から送られてくるデータが黒丸印のところしかなく、縦方向に 3 ライン間引いている状態を示している。モニターモードでは、このように CCD からは間引かれたデータしか出力されず、これをメモリに取り込むと、図 3 1 のように間引きによって上下方向につぶれた画像データとなる。図 3 2 のように、空間的に中心から同じ距離にあるデータを取り出しても、図 3 1 のように縦方向では横方向に対して画像が短く出

力されるという現象が現れる。この結果、[式1]による歪補正では、中心からの距離が遠いほど歪補正の効果が大きくなるようにしてあるので、縦と横で、歪補正の効果が変わってしまう。

このように縦と横で画像が非対称となるのを補正する目的で、補正係数 S_x 、 S_y が設けられている。図31の場合には、縦方向に伸ばす必要があるため、 $S_x = 1$ 、 $S_y = 4$ とする。

この S_x 、 S_y の効果により、縦、横のサンプリング比が異なっても正確な歪補正が可能となる。

また、前述のように、サンプリング比が整数でない場合は、 S_x 、 S_y も非整数で設定すればよい。さらに、撮像素子の各画素の間隔が縦横で異なる場合でも可能（例えば長方画素）である。

ところで、通常、JPEGですとYCbCrというデータで記録されるのが一般的であるが、そのときの設定として、図33及び図34に示すような場合が考えられる。例えば、YC422という形式のデータの場合は、輝度データに比べて色データについての解像度は横方向が1/2となっている。この場合は、色データについては $S_x = 2$ 、 $S_y = 1$ と設定して、補正処理を行えばよい。

ところで、歪の中心と画像の中心が一致しない場合、補正前に中心にあった被写体が、補正後は中心からずれてしまう。[式1]では、中心ずれ補正值(X_{off} 、 Y_{off})で補正することにより、補正前後で画像の中心に来る被写体が変わらないようにすることが可能である。

図35は、画像の中心ずれを説明する図である。

図35は補正画像を示す。前述の図4では、歪の中心と画像の中心が一致している、すなわち、CCDの中心（画像の中心）がレンズ光軸（歪の中心）に一致している場合であった。

図35は、CCDの中心に対して歪の中心がずれている場合である。図35では、歪中心 (X_d 、 Y_d) が画像の中心と異なるので、中心ずれを考慮せずに歪補正を行った場合、出力範囲Hに対する画像の中心PがQ点（大きな黒丸にて示

す) にずれる。ユーザにとっては、実際に撮影した画像と歪補正後に表示される画像の中心がずれると、非常に違和感を感じるものである。そこで、元々歪補正前に出力範囲Hの中心にあったデータを、出力範囲に対して中心の位置にくるようにするために、歪補正された画像データの切り出した方を変える。それを行うためには、画像の中心ずれを(X off, Y off)で定義して、画像データの切り出した方を変えればよい。画像の中心ずれ(X off, Y off)を設定することで出力範囲をHからH' に移動させ、元々画像の中心Pにあった被写体を、補正後の出力範囲H' においても中心にくるように補正することができる。

図35の実施例によれば、中央の被写体がずれることによる違和感を無くすことができる。

以上述べたように本発明の第2の実施の形態によれば、小領域（例えばブロックライン）の空間的な位置を算出可能とし、小領域単位での歪補正処理を実現することができる。また、間引き取り込みされた画像データに対しても、正確な歪補正を行える一方、デジタルズームなど、任意領域を切り出したデータに対しても正確な歪補正を行うことができる。

本発明の第3の実施の形態の画像処理装置の全体構成は、図21と同様である。

図37は、図2における歪補正処理部8の詳細な構成を示している。

図37の各部と図2の各部との対応関係を説明する。図37における補間位置算出回路22が図2の補間位置生成部811に、セレクタ24が図2のセレクタ813に、歪補正係数算出回路21及び補間位置補正回路23が図2の歪補正座標変換部812に、それぞれ対応している。また、図37における2ポートSRAM26が図2の内部メモリ部（内部バッファ）82に対応し、書込みアドレス生成回路28、バッファ空き容量監視回路29、データ送信可否判定回路30、バッファ開放量算出回路31及び読出しアドレス生成回路25が図2のメモリ制御部83に、補間回路27が図2の補間演算部84に、それぞれ対応している。エラー検出回路32は、図2には図示していないがメモリ制御部83に接続して設けられている。エラー検出回路32は、歪補正処理において歪量が大きくなり後述

する $preULB$ 、 $postULB$ の設定値を越える歪が発生したときに、エラー (ERROR) として CPU 4 に伝える機能を有するものである。

補間位置算出回路 22 は、歪補正処理部 8 の後段回路からのリクエストに応じて後段回路へグラント (リクエスト受付) を返したときにそのグラントをトリガとして、1 ユニットライン (以下、1 UL という) 分の補間位置 (X_1 , Y_1) を算出するものである。ここで、1 UL とは、前述したブロックライン処理において、メモリ部への書き込み、読み出しの際に、列方向に一行に並んだ一定数分の画像データを書き込んだり、読み出したりする際の一単位である。つまり、1 UL は、ブロックライン (BL) 上の列方向に一行に並んだ一定画素数のデータを指している。

補間位置補正回路 23 は、歪補正係数算出回路 21 からの歪補正係数 F を補間位置 (X_1 , Y_1) に乗算して、歪補正前の元データの座標位置 (X' , Y') を計算する。セレクタ 24 は、(X_1 , Y_1) と (X' , Y') を選択するもので、歪補正を行う場合は (X' , Y') を選択して出力し、拡大・縮小処理 (リサイズ) のみを行う場合は (X_1 , Y_1) を選択して出力する。

2 ポート SRAM 26 は、歪補正処理部 8 内におけるデータを格納するバッファである。

読出しアドレス生成回路 25 は、補間位置に対応する 2 ポート SRAM 16 内のアドレス (ADR) を生成したり、2 ポート SRAM 26 からの出力を整列するための制御信号を出力したり、出力画像データに同期して書き込み制御信号 WE_N を出力したり、図 38 及び [式 3] に示す D_0 が 2 ポート SRAM 上のどこかの位置かを知らせるデータ列制御信号を出力したりする。

書き込みアドレス生成回路 28 は、書き込み制御信号 WE に合わせて内部メモリである 2 ポート SRAM 26 のアドレス (ADDRESS) を生成したり、1 UL 分のデータ入力完了したら、その内部カウンタ (BLC) をカウントアップする。

データ送信可否判定回路 30 は、BLC 値、本回路の動作状態、次の UL 先頭

座標、歪補正処理部 8 の後段回路からのリクエスト (REQ) 状態とから、後段回路からの REQ 信号に対して Grant (GRANT_N) を送信できる状態にあるかを判定し、Grant 送信可能なら GRANT_N = 1 とする機能を有する。

補間回路 27 は、補間位置に対応する画像データにつき、16 点補間を行う。バッファ開放量算出回路 31 は、現在処理中の UL 先頭座標と、次に処理予定の UL 先頭座標の整数部の差をバッファ開放量 (図 50 参照) として算出する。

バッファ空き容量監視回路 29 は、内部バッファとしての 2 ポート SRAM 26 に格納されているデータ (UL) 量を保持し、バッファに空きがあれば、歪補正処理部 8 の前段回路に対してリクエスト (REQ = 1) を送信する。

次に、図 37 の歪補正処理部 8 の回路動作を説明する。

まず、内部メモリ部 (内部バッファ) である 2 ポート SRAM 26 に空きがある場合、バッファ空き容量監視回路 29 から前段回路に対してデータ要求としてのリクエスト (REQ) を送信すると、バッファ空き容量監視回路 29 は前段回路からリクエスト受付としての Grant (GRANT) を受信し、この受信と同時に 2 ポート SRAM 26 に格納可能な UL 数を記憶するカウンタ (回路 29 内にある) を 1 つ減らす。1 回のリクエストと Grant で 1 UL を一動作単位としてデータ転送する。上記カウンタが 0 になったらリクエストを取り下げる。

そして、前段回路から書込みアドレス生成回路 28 へデータが流入し、2 ポート SRAM 26 への書込みが行われる。1 UL 入力毎に書込みアドレス生成回路 18 の内部カウンタ (BLC) がアップする。

2 ポート SRAM 26 は、補間回路 27 にて例えば 16 点補間を行えるように、図 39 の如く読出しと書込みが同時に行える 2 ポート SRAM が 4×4 の計 16 個で構成されている。

ここで、図 39 を用いて、2 ポート SRAM 26 を説明するが、メモリの数や各メモリの大きさは異なっても構わない。例えば、図 39 では、ブロックラインの幅 (UL の長さ) を 96 としているが、もっと長くしても短くしても構わない。回路規模と補正性能とのバランスで決めればよい。また、 4×4 (計 16

個)も、4点補間であれば 2×2 (計4個)で構わない。補間方式(何点で補間するか)に合わせて決めればよい。

図39は、16個の2ポートSRAMからなるメモリ空間で、横方向は入力されるUL順であるが、縦方向は各2ポートSRAMごとに付したアドレスである。Nを整数とし、処理のある時点での状態を説明する。縦方向に並んだ4つの各2ポートSRAMであるNo.0, 4, 8, 12に格納するデータは、 $4N$, $4N+4$, $4N+8$, $4N+12$ 番目のULのデータであり、No.1, 5, 9, 13に格納するデータは、 $4N+1$, $4N+5$, $4N+9$, $4N+13$ 番目のULのデータであり、No.2, 6, 10, 14に格納するデータは、 $4N+2$, $4N+6$, $4N+10$, $4N+14$ 番目のULのデータであり、No.3, 7, 11, 15に格納するデータは、 $4N+3$, $4N+7$, $4N+11$, $4N+15$ 番目のULのデータである。

横方向に並んだ4つの各2ポートSRAMであるNo.0, 1, 2, 3及びNo.4, 5, 6, 7及びNo.8, 9, 10, 11及びNo.12, 13, 14, 15の各組についても同様に格納されている。

図40及び図41を使って、各2ポートSRAMにデータが書き込まれる様子を説明する。図40及び図41では、図40に示すように、幅8ライン(画素)のブロックラインを想定しており、データは、UL単位に「1, 2, ..., i, m」, 「5, 6, ..., j, n」, 「9, 10, ..., k, o」, ...という順序で入力される。図41は図40に示した書込み順番のデータが 4×4 の16個の2ポートSRAM No.0 ~ No.15上のどこに書き込まれていくかを示している。例えばNo.0, No.4, No.8, No.12には、図40の左側縦一列の1UL分の画像データ(順番1 ~ mで示されるデータ)が分散されて格納されていく。なお、図39に示される縦方向の2ポートSRAM No.0, No.4, No.8, No.12の各縦ラインごとに付された表記0 ~ 92, 1 ~ 93, 2 ~ 94, 3 ~ 95は、図40で示される各ユニットライン(1 ~ m ..., 5 ~ n ..., 9 ~ o ..., ...)の表記とは異なっている。これは、図40及び図41では、 4×4 の16個の画像データにつき書き込み順が分

かるように 1 ～ 16, a ～ p, A ～ P, …と 16 個の塊ごとに符号を付してあるためである。図 39 の場合はその縦方向の 2 ポート S R A M No.0, No.4, No.8, No.12 の各縦ラインごとに付された表記は、縦方向の 1 つの U L におけるデータ順に相当する。2 ポート S R A M No. 0, No. 4, No. 8, No. 12 のそれぞれには 1 つの U L 画素データ No.0 ～ 95 が交互に分散して記載されている。従って、図 39 の縦方向に並んだ 4 つの各縦ラインには 1 つのユニットライン (U L) の画素順 (0 ～ 95) の何番目の画素であるかを示す表記が付されている。図 39 の 2 ポート S R A M の No.0 の縦方向には U L の 4 の倍数の画素 (0, 4, 8 … 92) が格納してあり、2 ポート S R A M の No.4 の縦方向には U L の (4 の倍数) + 1 の画素 (1, 5, 9 … 93) が格納してあり、2 ポート S R A M の No.8 の縦方向には U L の (4 の倍数) + 2 の画素 (2, 6, 10 … 94) が格納してあり、2 ポート S R A M の No.12 の縦方向には U L の (4 の倍数) + 3 の画素 (3, 7, 11 … 95) が格納してある。さらに、2 ポート S R A M の No.1, 2, 3 及び No. 5, 6, 7 及び No.9, 10, 11 及び No.13, 14, 15 の各組についても同様に格納してある。

再び図 39 に戻って、1 U L 分のデータは縦方向に並ぶ 4 つの 2 ポート S R A M に 1 画素ずつ順に格納されている。最初の U L データは No. 0, 4, 8, 12 (図 39 の 4 N のライン) に書き込まれる。次の U L データは No. 1, 5, 9, 13 (図 39 の 4 N + 1 のライン) に書き込まれる。図 39 のバッファ形態では、歪補正処理によって補間位置がどのように移動しても、バッファ内の任意の座標周辺 16 点を 1 回のアクセスで同時に抽出することができる構成としてある。

データ送信可否判定回路 30 が後段回路からリクエスト (R E Q _ N) を受信し、次の U L データが送れる状態にあればリクエスト受付 (G R A N T _ N) を出力する。このリクエスト受付 (G R A N T _ N) は自分自身へのトリガとなり、補間位置算出回路 22 が動作を開始し、1 U L 分動作したら次の U L 先頭座標を算出して終了する。

補間位置算出回路 22 が動作開始して補間位置 (X_1, Y_1) を出力するのと同期するようにデータ送信可否判定回路 30 からは歪補正係数算出回路 21 へ動作トリガ (trig) を送る。歪補正係数算出回路 21 も補間位置算出回路 22 と同様、1UL 分動作したら次の UL 先頭座標を算出して終了する。

読み出しアドレス生成回路 25 は、入力される補間座標から、2 ポート SRAM 26 の 16 個夫々に読み出しアドレスを発行する。

次に、読み出し及び補間の方法について説明する。

補間方法を図 38 に示す。図 38 は補間回路 27 における補間演算のイメージ図である。補正座標位置 P の座標 (X', Y') は前出の [式 1] で既に求めている。この座標における画素値 (輝度データ) を求めるのに、座標 P (X', Y') の周辺 16 点の画素データ $D_0 \sim D_{15}$ から求める。その内、 D_0 が 16 個のメモリのどのメモリから来ているかが分かれば、 $D_1, D_2 \dots, D_{15}$ は D_0 に対する位置関係から分かる。後述するように、 D_0 は補間位置の座標によって求まる。

図 42 は 16 個の 2 ポート SRAM からなるバッファからの読出し例を示している。今、補間位置 (X', Y') が $X' = 10. \dots, Y' = 50. \dots$ (\dots は小数点以下の数字) とすると、 $X' = 10. \dots$ は $(4 \times 0 + 10)$ 以上であるから、横方向には 2 ポート SRAM の $4N + 10$ 番目の UL より若干右側に補間位置が来る。 $Y' = 50. \dots$ は $(4 \times 12 + 2)$ 以上であり、 $4N + 10$ 番目の UL で、50 番目の画素が格納されているメモリ (No. 10) が図 7 における画素 D_5 を出力するので、 D_0 を出力するメモリは、その左上にある No. 5 となる。

図 38 の $D_0 \sim D_{15}$ に対応する画素データは図 42 の○部であるので、それらが出力されるようにアドレスを生成する。

No. 0 から出力される画像データが D_0 には対応しない。図 10 の例では、No. 5 からの出力が D_0 に対応する。このため、どのメモリからどのデータが出力されているかを識別するため、読み出しアドレス生成回路 25 からデータ列制御信号を出力し、これによって、補間演算を行う補間回路 27 は D_0 が 2 ポート

SRAM 26 のどこから出ているかを認識して 16 点補間を行う。

画素データ $D_0 \sim D_{15}$ が分かれば、[式 4] の補間式による補間処理を行うことによって、補正座標位置の画素データを D_{out} として求めることができる。

[式 4]

$$\begin{aligned} D_{out} = & k_{x0} (k_{y0} D_0 + k_{y1} D_4 + k_{y2} D_8 + k_{y3} D_{12}) \\ & + k_{x1} (k_{y0} D_1 + k_{y1} D_5 + k_{y2} D_9 + k_{y3} D_{13}) \\ & + k_{x2} (k_{y0} D_2 + k_{y1} D_6 + k_{y2} D_{10} + k_{y3} D_{14}) \\ & + k_{x3} (k_{y0} D_3 + k_{y1} D_7 + k_{y2} D_{11} + k_{y3} D_{15}) \end{aligned}$$

最後の UL データ出力の後、バッファ開放量算出回路 31 は今処理し終わった UL 先頭座標と次の UL 先頭座標との差を計算し (図 50 参照)、必要なくなったデータの蓄積されているバッファを開放するため、バッファ開放量をバッファ空き容量監視回路 29 へ出力する。ただし、歪中心を跨ぐ時のバッファ開放量は、図 49 に示すように $postULB1$, $postULB2$ の値変化も考慮した量を開放することが好ましい。図 48 に示す $preULB$ 、 $postULB$ とは、UL を形成する画素のうち、最初の画素の座標位置に対して、行方向に対して前側及び後側に所定幅の領域を設けたもので、これらをそれぞれ $preULB$ 、 $postULB$ と定義している。歪中心を跨いだ時はバッファ開放量を通常の UL 先頭座標の差ではなく、参照値の変わる $postULB$ (ULB: Unit Line Buffer の略、ユニットラインバッファ) の変化量を参照してバッファ開放量を調節する (調整量は、 $postULB1 - postULB2$ で、図 49 参照)。

(UL 先頭座標の差) + (調整量) がマイナスになる場合は、マイナス値を記憶しておく。処理が進んでバッファ開放量がマイナス値を超えるまでバッファの開放を行わない。

前述のバッファ開放量の算出と共に、バッファ開放量算出回路 31 は次の UL 処理にあとどれだけ前段回路からデータが必要かをデータ送信可否判定回路 30 へ送信する。

バッファ空き容量監視回路 29 は前述のバッファ開放量の算出によってバッフ

ァに空きができたなら、前述の前段回路へのリクエストを行う。

データ送信可否判定回路30は書込みアドレス生成回路28の内部カウンタ(BLC)とバッファ開放量算出回路31からの入力とpreULB値とに基づいて、次のULデータを送れるかどうかを判断する。データ送信可否判定回路30は、後段回路のリクエストに対してリクエスト受付(GRANT_N)を返す。

エラー検出回路32は、読出しアドレス生成回路25に入力される座標がブロックライン(BL)の左端(図43参照)や右端を逸脱したり、ブロックライン(BL)の上下端(図44参照)を逸脱したり、歪量がpreULB、postULBの設定値を逸脱(図45参照)した場合にエラーを出力する。これらのエラーの出力は、座標を判別して出力することになる。図43及び図44の場合には、エラーを出力するが、処理は継続する。図45の場合も、エラーを出力するが、処理は継続する。図45のpreULB、postULBの場合については、エラーを出力するだけではなく、逸脱した量を記憶し、レジスタに設定してCPU4がデータとして取得できるようにしておく。

このように、ブロックライン処理で、出力側の画像に対して入力側の画像を補間算出しているときに入力画像の範囲からはみ出すようなときがある。その場合、入力範囲にデータがない部分で補間データを生成することになり、エラー検出回路32から補間ができない旨のエラーを出力することになる。

図46にあるように、歪の中心座標に対して、画像の歪み方は対照的になる。また、歪みは一般的に中心に近いほど小さく、離れるほど大きくなる。そこで、BLの両端にあるULの変形を考慮し、バッファを確保すれば、全てのULにおいて、補間処理でデータが不足することは通常はない。そこで、補間する最初の座標(図ではブロックラインBL内の画像データの一番上の×印にて示す先頭座標)に対して前方(図示右側)に所定の領域preULBを、後方(図示左側)に所定の領域postULBを設ける。

サポート機能は、図1の「歪補正範囲算出部」の部分に該当する。図12において、注目領域がBLとなるが、歪補正の座標変換で図13のように変形される。

このとき、 X'_{TL} , Y'_{TL} , X_{Lmin} , X_{Lmax} から $preULB1/postULB1$ が決められる (図 48 参照)。同様に、 X'_{TR} , Y'_{TR} , X_{Rmin} , X_{Rmax} から $preULB2/postULB2$ が決められる。

さらに、UL 処理において、歪の中心を通過すると、光学的歪の曲がり方が逆方向となる。すると、図 46 のように、歪中心の左右では $preULB$ 、 $postULB$ の値が異なる。この値を、左右どちらにも対応できるように、図 47 のように右側で最も大きい $postULB$ を採用し、左側で最も大きい $preULB$ を採用すると、左側又は右側の一方側の処理のときに必要もないのにもう一方側のデータを大きく (すなわち、バッファを広く) 確保しておかなければならず、バッファの無駄である。

そこで、 $preULB$ と $postULB$ で予め同じ大きな値で両方決めておくのではなくて、 $preULB$ と $postULB$ を変数としてそれぞれに値を入れるようにする。歪の中心座標を過ぎたら、これらの値を変えてやって確保量を変えるようにする。すなわち、歪中心前後で $preULB$ 、 $postULB$ を変化することにより、内部バッファの使い方に無駄をなくし、小さなバッファで比較的大きな歪補正を行える。

pre 、 $postULB$ をレジスタ 85 に設定するが、それらの値を生成するためには、歪の変形を考慮して画像データの入力範囲を算出可能とするサポート機能が必要となる。このサポート機能は、歪補正処理部 8 の歪補正処理機能に付加するなどして設ければよい。メモリ制御部 83 は、1 UL 処理中は他の処理によって前記領域 $preULB$ 、 $postULB$ が上書きされないように制御する。前記領域 $preULB$ 、 $postULB$ の設定は、CPU 4 からレジスタ 85 (図 2 参照) に設定しても構わないし、CPU で自動的に計算をして設定してもよい。

歪補正処理において 16 点補間すると前に述べたが、図 48 に示すように、歪補正によって補間位置は 1 UL 処理の間に a (ただし、 $a > 0$) だけ動き得る。

16 点補間する関係上、入力画像範囲としては pre 側、 $post$ 側とも上記間隔 a に、補間に必要な画素分の間隔 b_L 、 b_R を加えた範囲が必要となる。

歪中心を越えると、pre, postULBの値を変えることで、バッファを必要最小限に絞る。また、前述の「歪補正算出部」で、補間分を考慮して結果を出力させてもよい。

歪補正処理部8の持っている内部メモリ部（内部バッファ）82は、図39の場合、最大で縦96画素の横に16ライン（画素）である。これを使って歪補正していくことになる。ブロックライン処理のデータは少なくとも1ULずつ図示右方向にスワイプしていく。一気に数UL開放されることもあるので、必ずしも1ULずつスワイプするわけでない。内部メモリ部82は、はじめは空の状態からスタートするので、16ライン分入ってきて、歪補正処理を行う。これらのユニットラインを処理していくと、左側のラインに不要なデータ（1から数UL分のデータ）が生じることになる。なお、処理したら必ず不要になる訳ではなく、拡大率が高い時は不要になるまで数ULかかることもある。不要なデータ分はバッファを開放（すなわち、上書き許可）していった新しいデータを入れていくことにする。スワイプしていく画像は右側にずれていく。最大16ライン読み込めるサイズしかないの、要らなくなったデータ領域は開放して新しいデータを順次上書きしていくようにする。不要となったバッファ領域は全て一気に開放してしまう。その開放量は1ラインであるかもしれないし、5ラインであるかもしれない。例えば、はじめの3ライン分のデータがバッファに入っていたものを以降の処理に不要であるので開放してしまっ次データを受け付けるようにして、次のデータが上書きされる。

ところで、1UL処理が終わった後に開放できるバッファの量は、次のUL先頭座標が決まらなければ分からない（図50参照）。内部バッファは小さいので、使わなくなったデータが格納されている部分をできるだけ早くに開放する為、次のUL先頭座標を算出して開放量を取得し、バッファを開放することで内部バッファを有効に利用できるようにする。ところが、歪補正に対して後段の回路ブロックからリクエストが来てそれに対するグラントを返して1ULのデータを後段へ出力し、また次のリクエストが来てからグラントを返した後、座標を求めてと

いうように動作させると、バッファの開放が遅れ、これに伴って、新しいデータの取り込みも遅れ、結果として、パイプライン動作に長い空きができてしまうことが起こり得る。そこで、図5 1に示すように先の歪補正処理（図示の座標1～ n の処理）を行っている段階で、後段の回路ブロックからリクエストに関係なく、座標 n の処理が終わったら次の先頭座標 $n+1$ を演算して開放量を求め、バッファを早期に開放してしまう。これにより、前段回路ブロックから新たなデータ入力が比較的早く行なわれ、パイプラインの空きを詰めることができる。しかしながら、図5 1の場合であると、座標 n の処理をした後、画素 $n+1$ の座標を計算してから処理を終了するので、図5 2のようにバッファ内に次のUL処理に必要なデータが揃っているのに、後段からのリクエストが連続するとパイプラインに空きが生じる。依然としてパイプラインの空きは存在するので、リクエストが連続した時は、装置全体の動作速度に悪影響を及ぼす恐れがある。

そこで、図5 1の改善策として、図5 3のように、なるべく早くに次の座標が決まってバッファの開放量を求める方法がある。図5 1のように一番最後に次の先頭座標を求めるよりももっと早い段階で次のUL先頭座標を取得しておき、必要なデータが揃っていると判断されたら、次のUL処理を連続して行うことで処理の空きを小さくする。すなわち、1 UL処理が終了する前にULのデータをレジスタ等に保持しておいて、次のULの先頭の座標位置を算出することで、予め次のUL処理の開放量を求めておく。図5 3では、先の歪補正処理（図示の座標1～ n の処理）の2画素目に次のULの先頭座標を求めておいてバッファ開放量を早い段階で取得する。これによって、バッファの開放を早くすることができ、図5 4に示すように図5 2に比べてパイプラインの空きをかなり少なくできる。しかしながら、回路的に複雑となり、制御も困難となる問題点がある。

そこで、図5 3をさらに改善したのが、図5 5である。図5 5では、現在処理中のULが終わった時に開放する量（開放量1）は前のUL処理中に既に分かっている。座標 n を算出した後、2つ先のUL先頭座標を計算し、次のUL処理が終わった時に開放できる量（開放量2）を予め求めておく。このようにすると、

図53の場合のようにUL処理中に例外的な座標生成（座標2の生成）を行う必要がない。このように、次のULの先頭ではなくて次の次の2つ先のULの先頭座標を求めておけば、図53のような例外的な処理がなくなるので、回路的には簡単になる。

以上述べたように本発明の第3の実施の形態によれば、バスの転送量やメモリの容量を大きく増大させることなく、歪補正を実現することができる。すなわち、小さなバッファ容量で比較的大きな歪補正処理を行うことが可能となる。

産業上の利用可能性

デジタルカメラ等の電子的撮像装置に用いられる画像処理装置、画像処理方法及び歪補正方法に広く利用可能であり、回路規模やデータ転送量を増大させることなく、歪補正機能を実現し、歪補正処理を行う場合に、空間的な位置を算出し、元データである撮影画像データを有効に活用した歪補正処理を行え、糸巻き型歪、樽型歪、及び陣笠型歪などの歪曲収差に基づく歪に対しても有効な歪補正処理を行うことができる。

請 求 の 範 囲

1. 歪補正手段を有する画像処理装置であって、

該画像処理装置は、さらに、

前記歪補正手段が歪補正処理を行う入力画像範囲を算出する歪補正範囲算出部を有することを特徴とする画像処理装置。

2. 前記歪補正範囲算出部は、

補間座標を生成する座標生成部と、

前記生成された補間座標に対して所定の歪補正式を適用して変換した座標を出力する歪補正座標変換部と、

前記変換された座標位置から前記入力画像範囲を算出する補正範囲検出部と、を有することを特徴とする請求項 1 に記載の画像処理装置。

3. 前記座標生成部は、歪補正後の出力画像範囲における辺縁部の各画素に対応する座標のみを用いて補間座標を生成することを特徴とする請求項 2 に記載の画像処理装置。

4. 前記出力画像範囲は矩形であって、前記座標生成部は、該出力画像範囲の 4 辺の各画素に対応する座標のみを用いて補間座標を生成することを特徴とする請求項 3 に記載の画像処理装置。

5. 前記歪補正範囲算出部は、座標変換により生成された座標に対し、前記出力画像範囲の 4 辺の各辺に対応する画素の座標の最大値と最小値および前記出力画像範囲の 4 頂点に対応する座標の少なくともいずれかから前記入力画像範囲を算出することを特徴とする請求項 1 ～ 4 のいずれか 1 つに記載の画像処理装置。

6. 前記歪補正座標変換部は、前記所定の補正式に含まれる演算を時系列に処理することを特徴とする請求項 2 に記載の画像処理装置。

7. 前記座標生成部は、前記歪補正処理のための補間座標に対して所定の間引きを行って座標を求めることを特徴とする請求項 2 に記載の画像処理装置。

8. 前記歪補正範囲算出部は、歪補正処理を行う複数入力信号に対して順次範囲

算出を繰り返して前記入力画像範囲を算出することを特徴とする請求項 1 に記載の画像処理装置。

9. 前記範囲算出を繰り返し処理で行い、入力画像範囲に対して歪補正後の画像範囲が所定の範囲になるような補正倍率 M を決定することを特徴とする請求項 1 又は 8 に記載の画像処理装置。

10. 前記歪補正範囲算出部は、

前記歪補正処理部が歪補正実行中に、次に歪補正を行う画像入力範囲を算出することを特徴とする請求項 1 に記載の画像処理装置。

11. 歪補正処理を行う画像処理方法であって、

前記歪補正処理を行うに際して、歪補正処理を行う入力画像範囲を算出することを特徴とする画像処理方法。

12. 画像データに対して歪補正処理を行う歪補正手段を有する画像処理装置において、

前記歪補正手段は、

補正前の画像データ内において、補正画像の各画素が対応する位置を所定の補正式に従って算出する際、撮像面における空間的な位置を記述可能な座標系において算出することを特徴とする画像処理装置。

13. 前記歪補正手段は、

補正画像に含まれる第 1 の小領域を単位に処理を行い、

前記第 1 の小領域の各画素の座標位置を前記補正式に従って変換した小領域を含む撮像面における第 2 の小領域について、前記撮像面に対応する座標系での位置を、前記第 2 の小領域での座標に変換した後に、

前記補正画像の各画素のデータを生成することを特徴とする請求項 1 に記載の画像処理装置。

14. 前記画像データは、

撮像データの一部のみを取り込んだ部分画像データであることを特徴とする請求項 12 又は 13 に記載の画像処理装置。

15. 前記部分画像データは、

前記撮像データの一部を切り出したデータであることを特徴とする請求項14記載の画像処理装置。

16. 前記部分画像データは、

前記撮像データの横または縦の少なくとも一方向に間引かれたデータであることを特徴とする請求項14記載の画像処理装置。

17. 前記画像データは、

前記撮像データに対してフィルタリング、空間サンプリングまたは補間処理の少なくともいずれか1つを行って生成されるデータであることを特徴とする請求項12記載の画像処理装置。

18. 前記画像データは、

各画素が撮像面で空間的に位置する間隔が縦方向及び横方向で異なることを特徴とする請求項12に記載の画像処理装置。

19. 画像データにおける各画素の空間的なサンプリング間隔が縦横で異なる場合、歪補正前の画像内における座標を求める補正式は、縦横でのサンプリング間隔の違いを補正する係数を含むことを特徴とする請求項12～18のいずれか1つに記載の画像処理装置。

20. 歪補正前の画像内における座標を求める補正式は、

所定量のオフセット補正が可能であることを特徴とする請求項12～19のいずれか1つに記載の画像処理装置。

21. 前記オフセット補正は、

歪の中心と画像の中心が一致しない場合に行うことを特徴とする請求項20に記載の画像処理装置。

22. 画像データに対して歪補正処理を行う歪補正手段を有する画像処理装置の歪補正方法であって、

前記歪補正方法は、

補正画像の各画素に対応する座標系での第1の位置を生成するステップと、

前記第 1 の位置を、歪補正式に従って歪補正前の画像データ内における座標系での第 2 の位置に変換するステップと、

前記歪補正前の画像データ内における座標系での第 2 の位置を、撮像面に対応する座標系を基準とする設定領域内での座標に変換するステップと、

前記設定領域内での座標に基づいて補間処理を行い、前記補正画像の各画素のデータを生成するステップと、

を有することを特徴とする歪補正方法。

23. 前記請求項 22 記載の歪補正方法においては、前記補正画像に含まれる小領域を単位に処理を行うことを特徴とする歪補正方法。

24. 画像データに対して歪補正処理を行う歪補正手段を有する画像処理装置において、

前記歪補正手段は、

前記画像データの一部を格納するメモリ部と、

前記メモリ部に対するデータの書き込みおよび読み込みを制御するメモリ制御部と、

を有し、

前記メモリ部から読み出した画像データに対する補間演算を行うことを特徴とする画像処理装置。

25. 前記メモリ制御部は、

前記メモリ部への書き込みの際は、列方向に一系列に並んだ一定数分の画素からなる画像データ（ユニットライン：UL）を単位として書き込むように制御し、前記メモリ部からの読み出しの際は、歪補正処理後の画像が前記ULを単位として出力されるように、前記メモリ部に格納された画像データに対して読み出し制御することを特徴とする請求項 24 に記載の画像処理装置。

26. 前記メモリ制御部は、前記ULの最初に処理される画素の座標位置に対して、行方向に対して前側および後側に所定幅の領域（それぞれpreULB、postULB）を設け、前記ULの処理中は他の処理によって前記領域が上書き

されないよう制御することを特徴とする請求項 25 に記載の画像処理装置。

27. 前記メモリ制御部は、さらに、

前記ULの処理中に、前記歪補正手段の前段からのデータを入力するために開放可能なバッファ量を算出するバッファ開放量算出回路と、

前記歪補正手段の後段に対するデータ送信の可否を判定するデータ送信可否判定回路とを有し、

前記バッファ開放量算出回路には、前記ULの処理における最初の画素に対する補間座標を中心として前記postULBが入力され、データ送信可否判定回路には前記preULBが入力され、バッファ開放量算出回路は前記バッファ開放量を前記postULBの値を参照して算出し、データ送信可否判定回路は前記preULBの値を参照してデータ送信可否の判定を行うことを特徴とする請求項 25 又は 26 に記載の画像処理装置。

28. 前記preULB値、postULB値は、それぞれ歪中心に対して前側と後側とで異なる値を設定可能であることを特徴とする請求項 26 又は 27 に記載の画像処理装置。

29. 前記画像処理装置は、さらにバッファの空き領域を検出するバッファ空き容量監視回路を有し、

該バッファ空き容量監視回路によりバッファ内の空き領域が検出されると、該空き領域へのデータの書き込みを可能とすることを特徴とする請求項 24～27 のいずれか 1 つに記載の画像処理装置。

30. 前記メモリ部は、データのリード動作とライト動作が同時に可能な複数のメモリからなり、

前記メモリ制御部は、さらに、

前記メモリ部へのデータ書き込み制御を行う書き込みアドレス生成回路と、

前記メモリ部に格納されている画像データから補間演算に必要なデータを同時に読み出すためのアドレスを生成する読出しアドレス生成回路と、
を有し、

前記データ書き込み制御は、同時に読み出すデータをそれぞれ異なるメモリに書き込むものであることを特徴とする請求項 2 4 記載の画像処理装置。

3 1. 前記画像処理装置は、さらに、歪補正処理された座標が前記 U L の処理における最初の画像に対する補間座標に対し、列方向に前記 p r e U L B、p o s t U L B を逸脱する座標が生成された時、若しくは、歪補正された座標が前段から入力されない座標となった時にエラーを検出して出力するエラー検出回路を有することを特徴とする請求項 2 6 ~ 2 8 のいずれか 1 つに記載の画像処理装置。

3 2. 前記エラー出力がなされたときはレジスタを再設定して画像処理を再実行することが可能であることを特徴とする請求項 2 6 ~ 2 8 のいずれか 1 つに記載の画像処理装置。

3 3. 1 U L の処理終了後、次の 1 U L またはさらにその次の 1 U L の処理を開始する際の最初の補間座標が演算された後に 1 U L 単位の処理を終了することを特徴とする請求項 2 5 ~ 2 9 のいずれか 1 つに記載の画像処理装置。

3 4. 画像データに対して歪補正処理を行う画像処理方法において、

前記歪補正処理を行うに際は、データの書き込みおよび読み込みが制御されるメモリ部に、前記画像データの一部を格納し、該メモリ部から読み出した画像データに対して補間演算を行うことを特徴とする画像処理方法。

3 5. 前記画像処理装置は、さらに、前記歪補正手段が歪補正処理を行う入力画像範囲を算出する歪補正範囲算出部を有すること、
を特徴とする請求項 2 6 ~ 2 8 のいずれか 1 つに記載の画像処理装置。

3 6. 前記歪補正範囲算出部は、所定の歪補正式を適用して座標変換可能であり、歪補正後の出力画像範囲において、

周辺 4 辺の各辺に対応する画素を座標変換して生成した変換座標の最大値と最小値、および 4 頂点に対応する画素を座標変換して生成した変換座標、の少なくともいずれかを算出し、前記 p r e U L B および p o s t U L B は、前記歪補正範囲算出部の出力結果から算出すること、を特徴とする請求項 3 5 記載の画像処理装置。

図 1

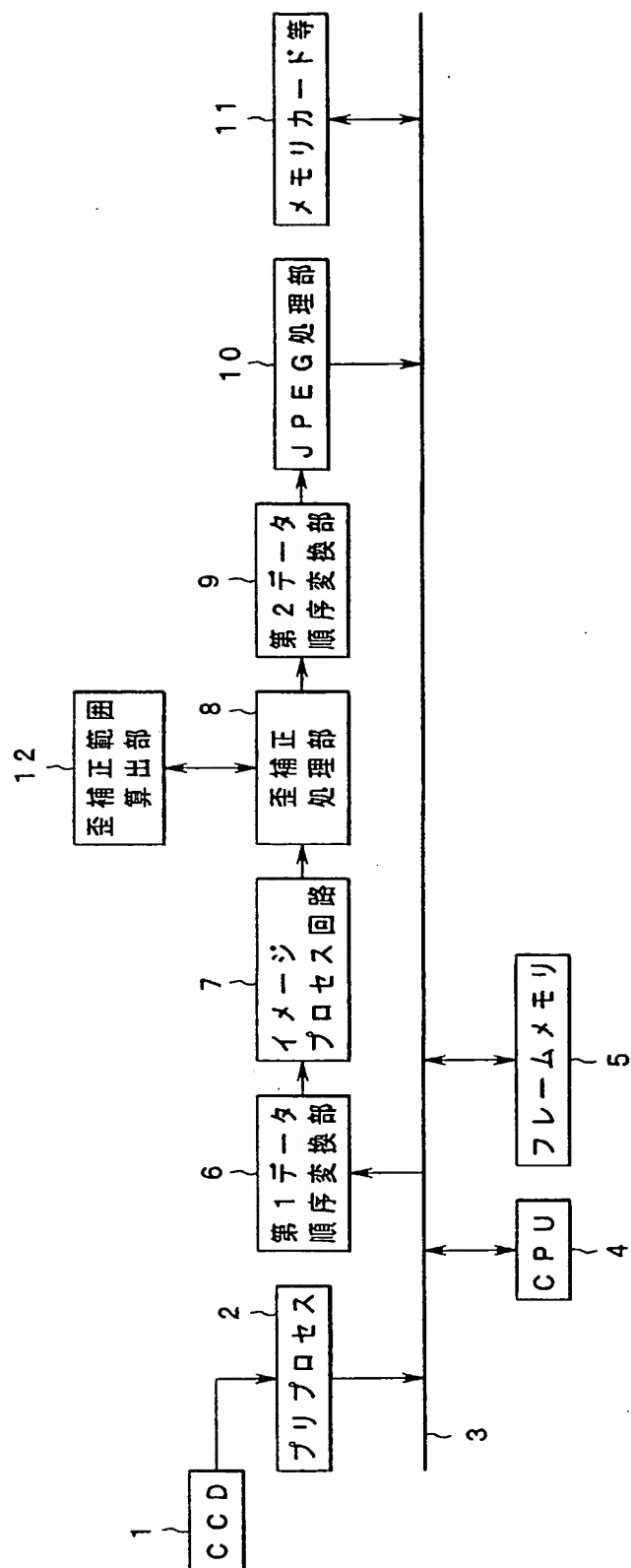


図 2

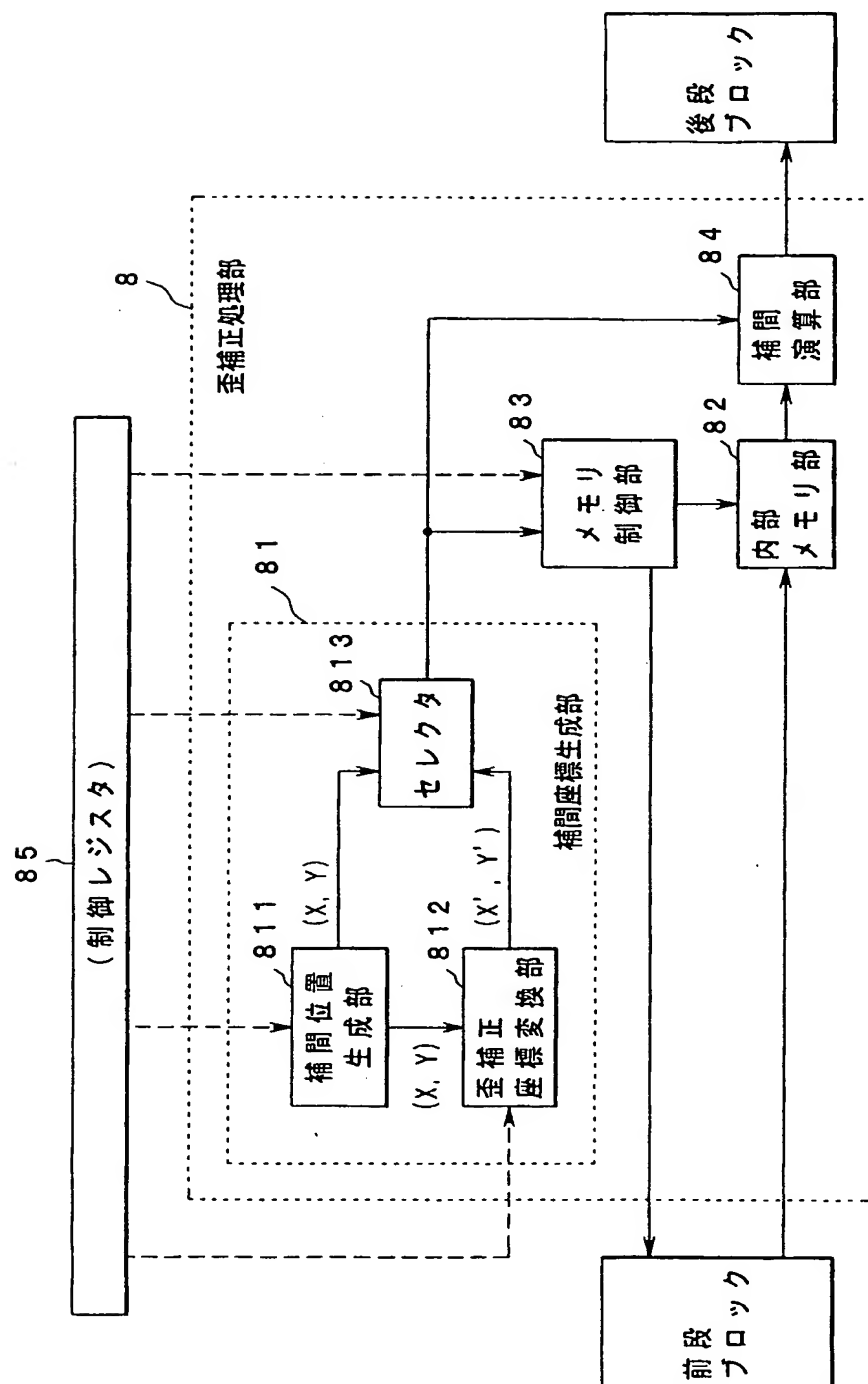


図 3

図 4

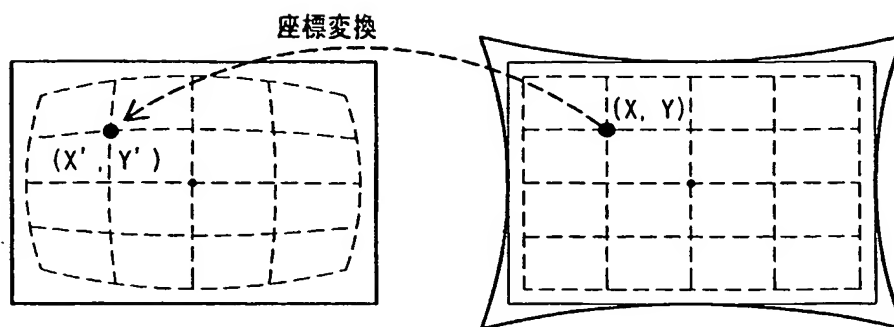


図 5

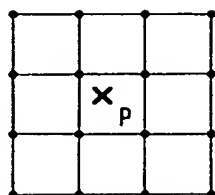


図 6

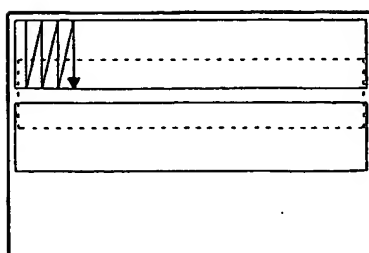


図 7

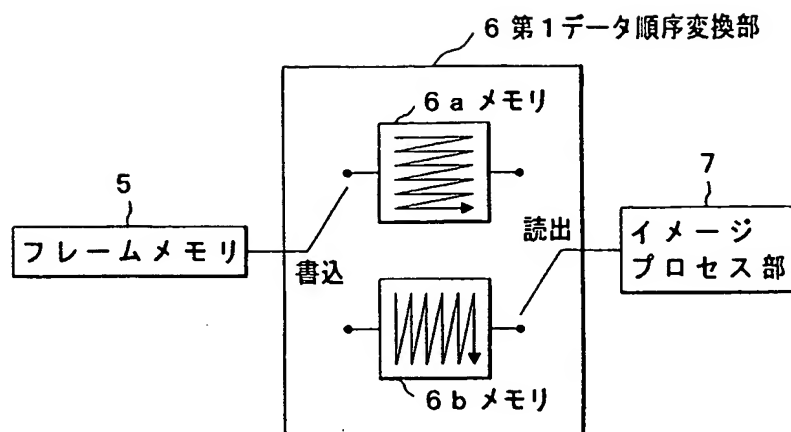


図 8

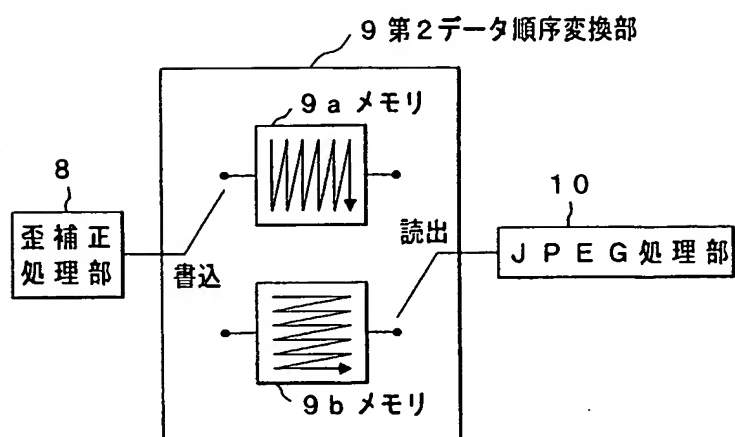


図 9

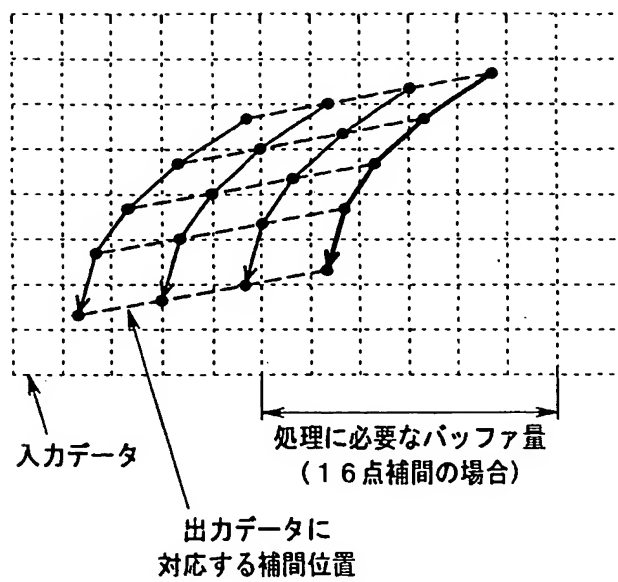


図 10

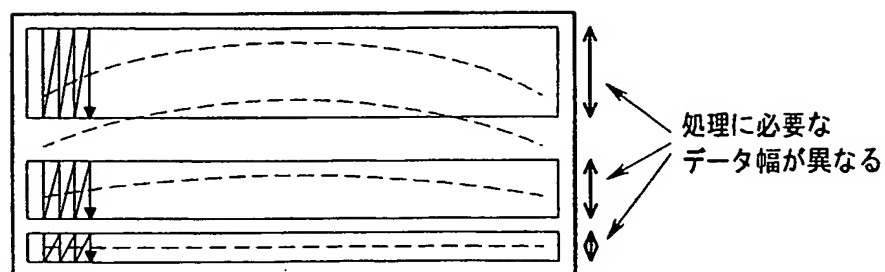


図 1 1

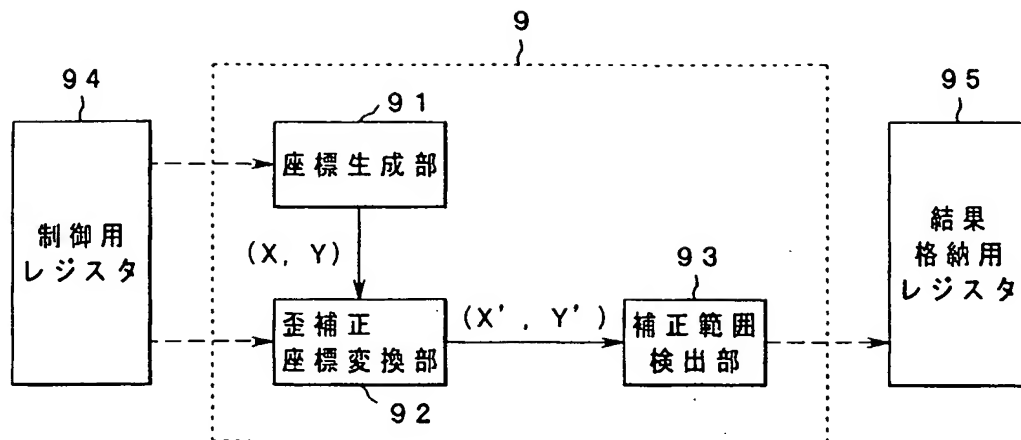


図 1 2

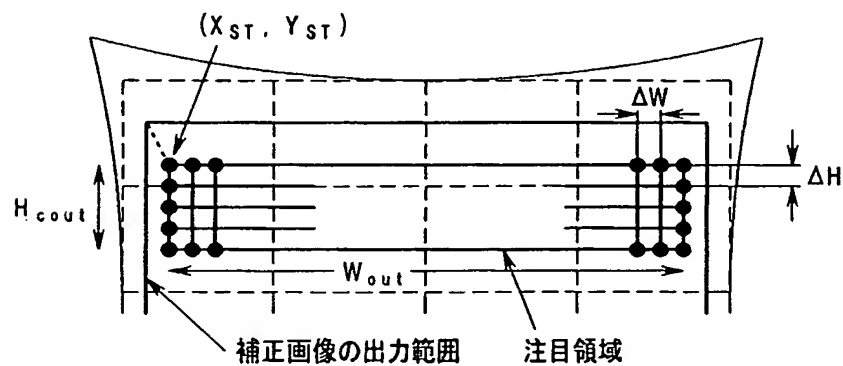


図 1 3

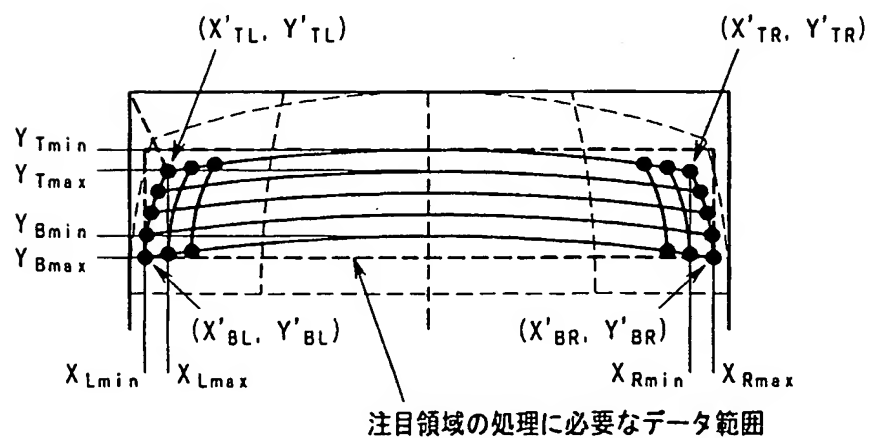


図 1 4

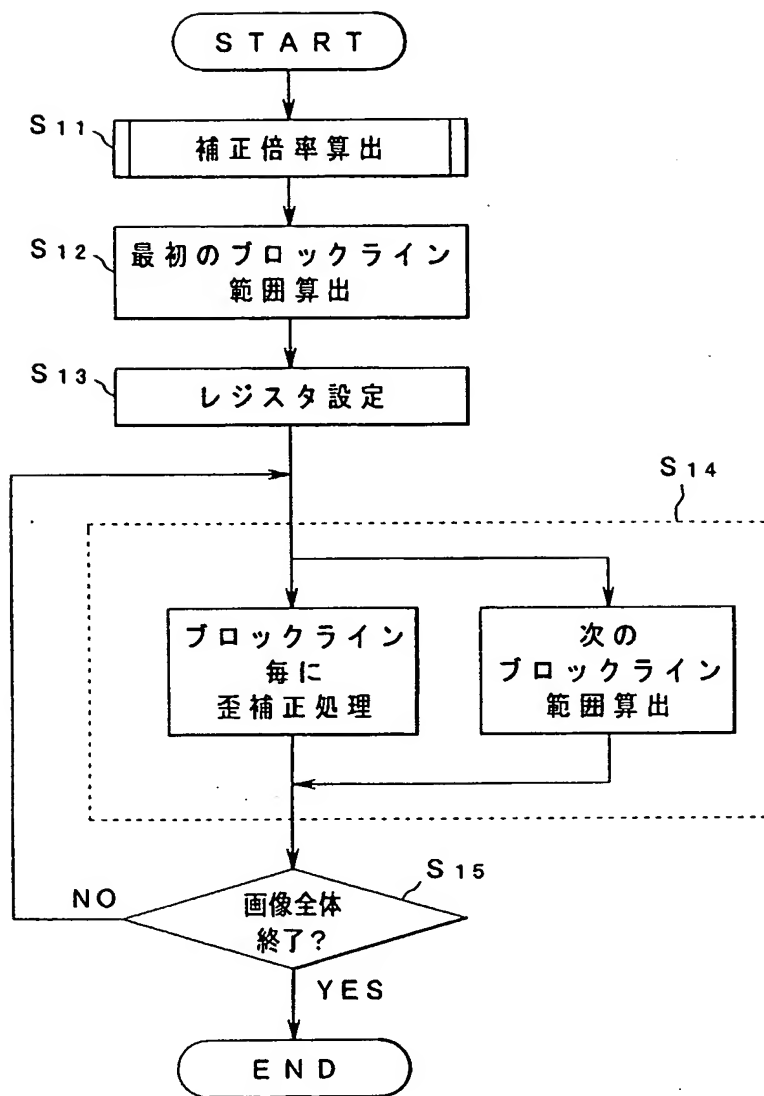


図 15

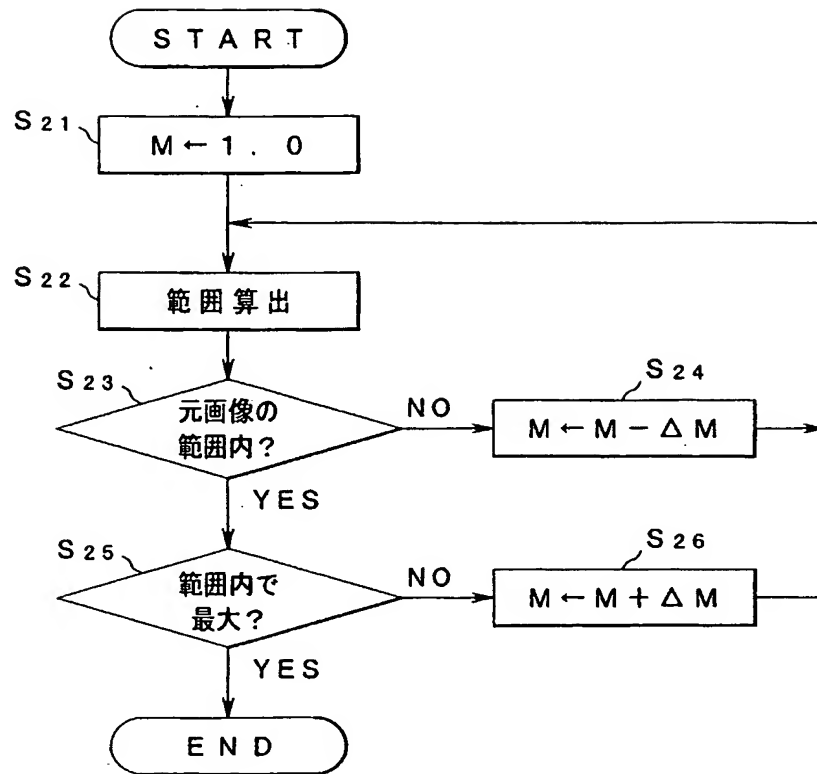


図 16

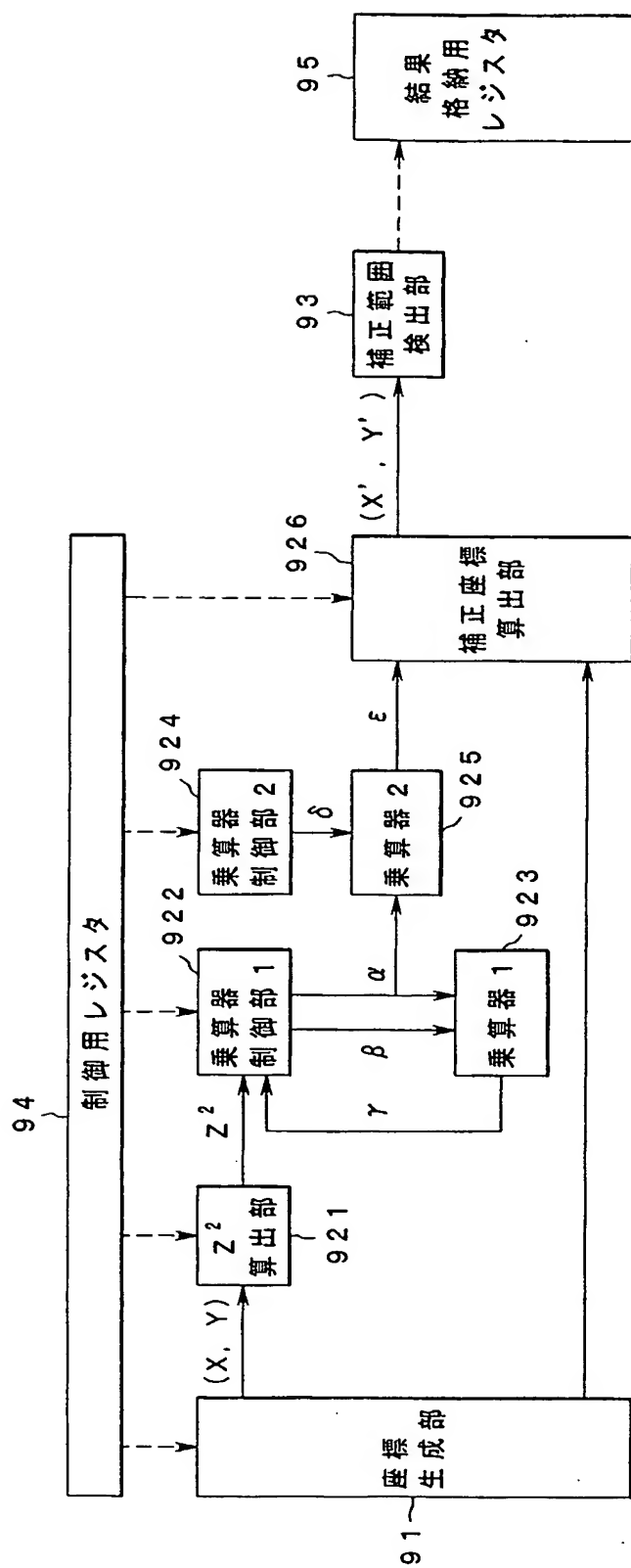
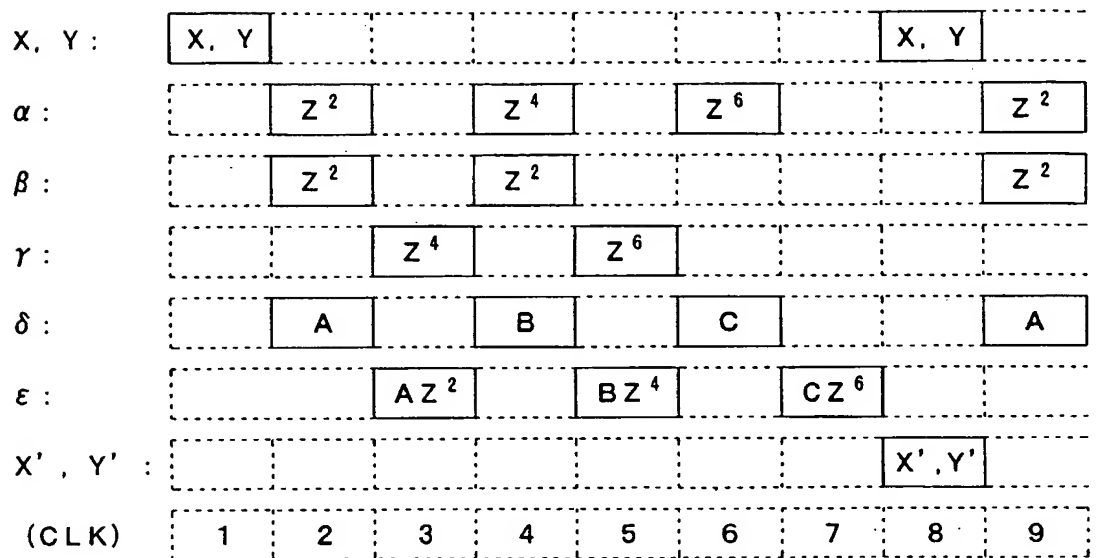
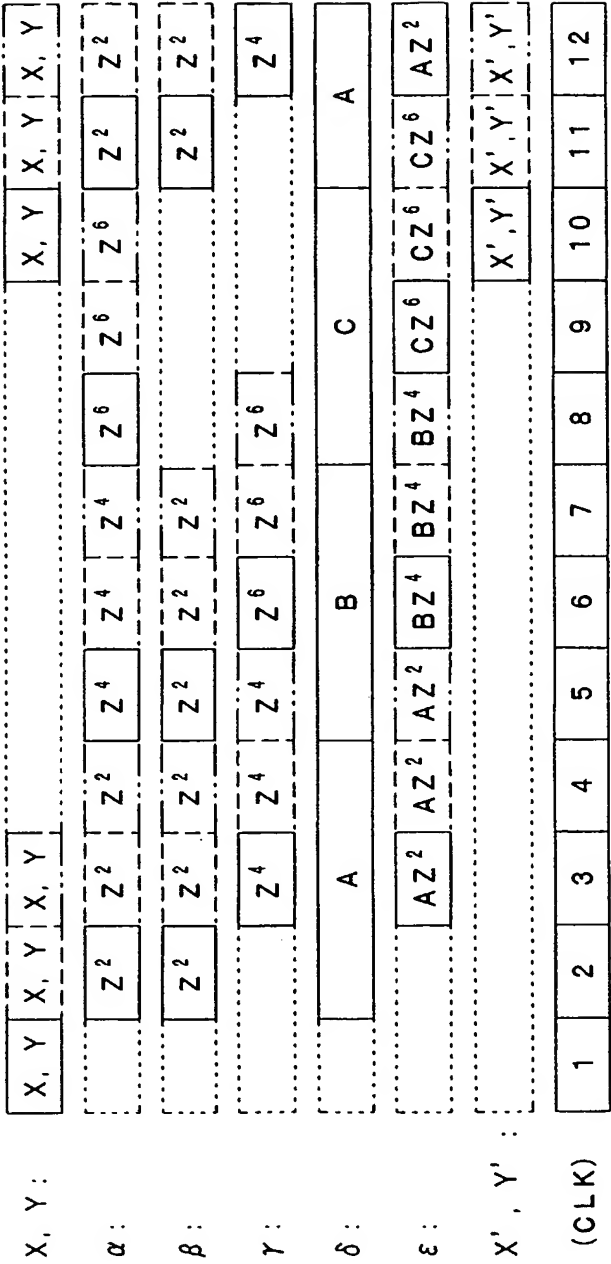


図 17



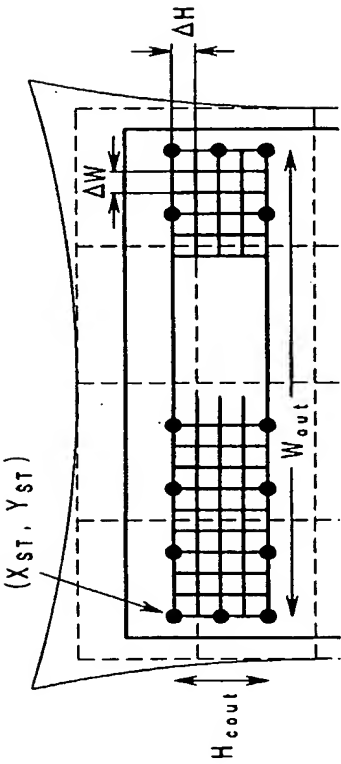
→ 時間

図 18



時間 →

図 19



20

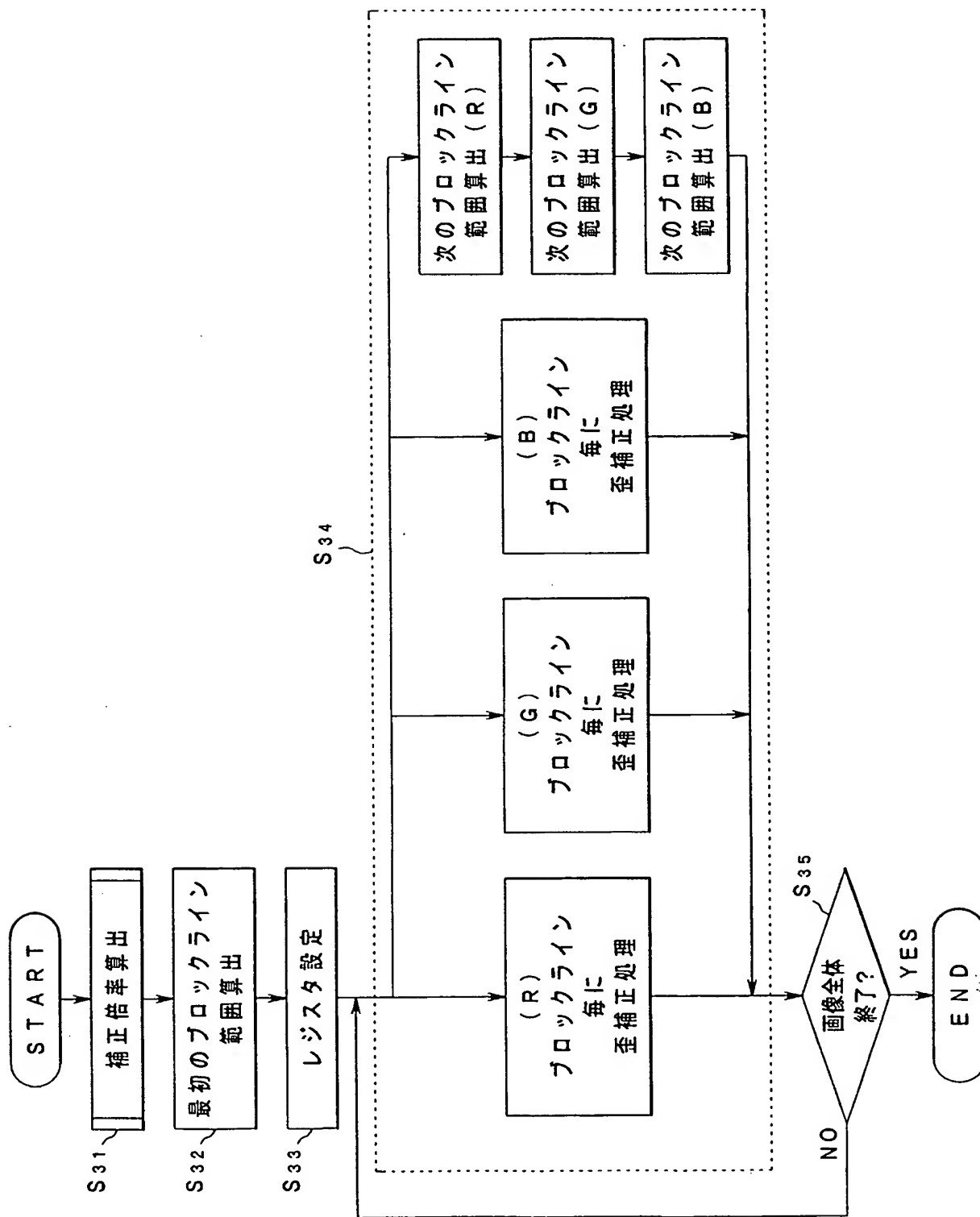


図 21

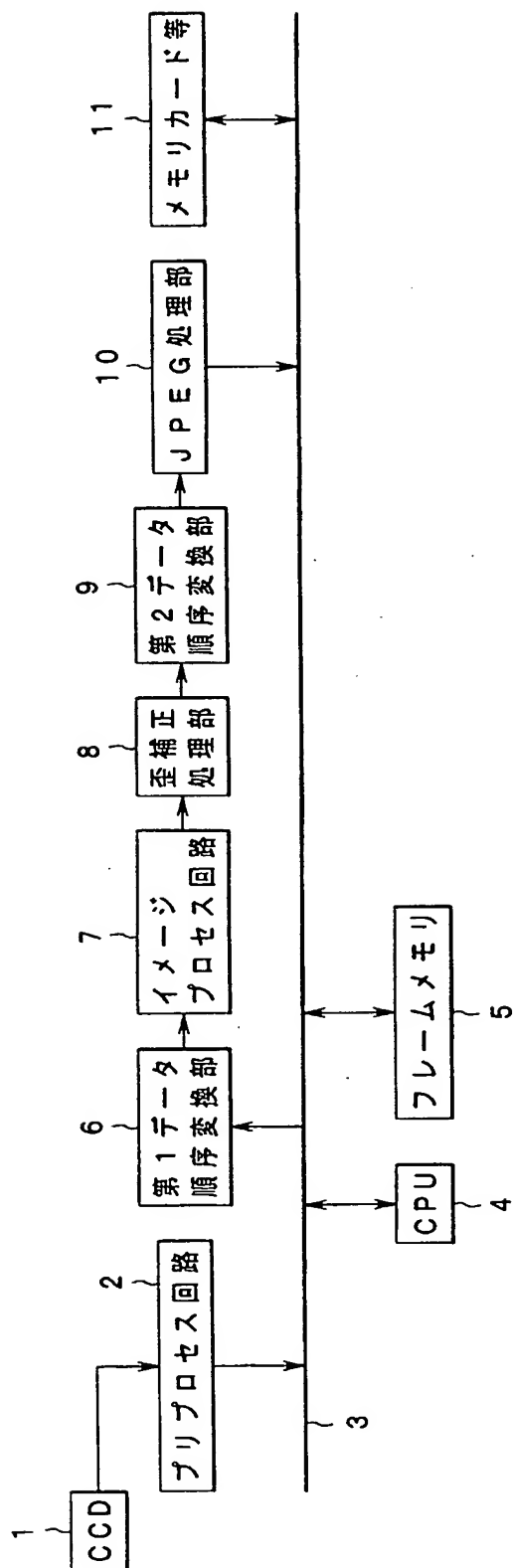


図 2 2

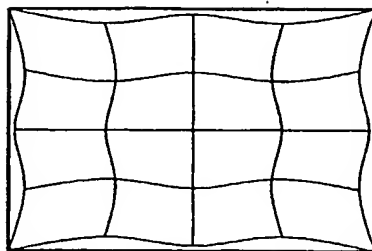


図 2 3

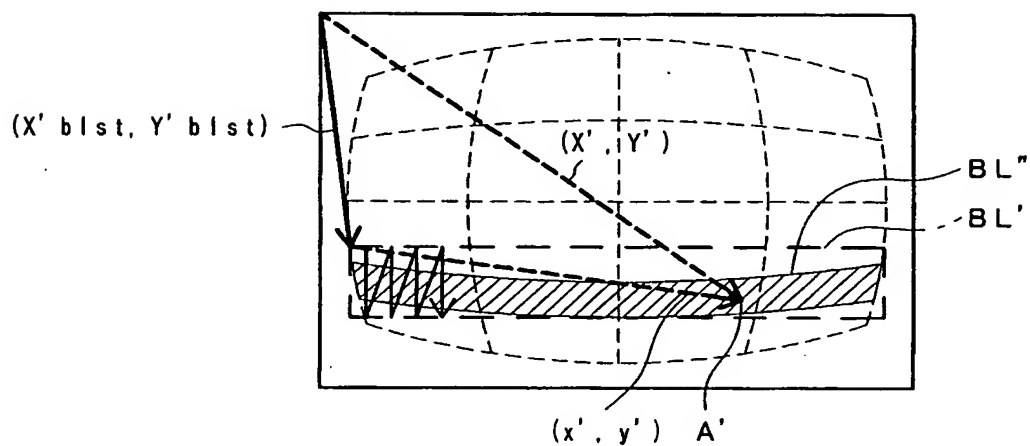


図 2 4

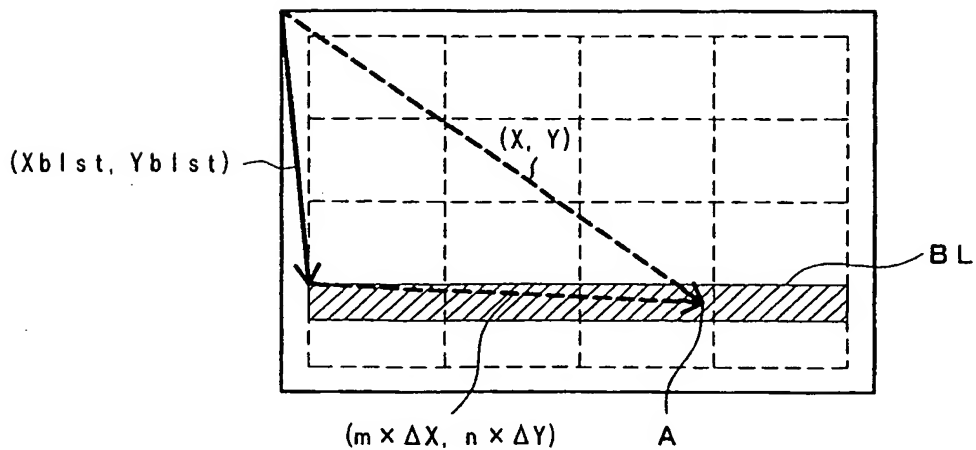


図 2 5

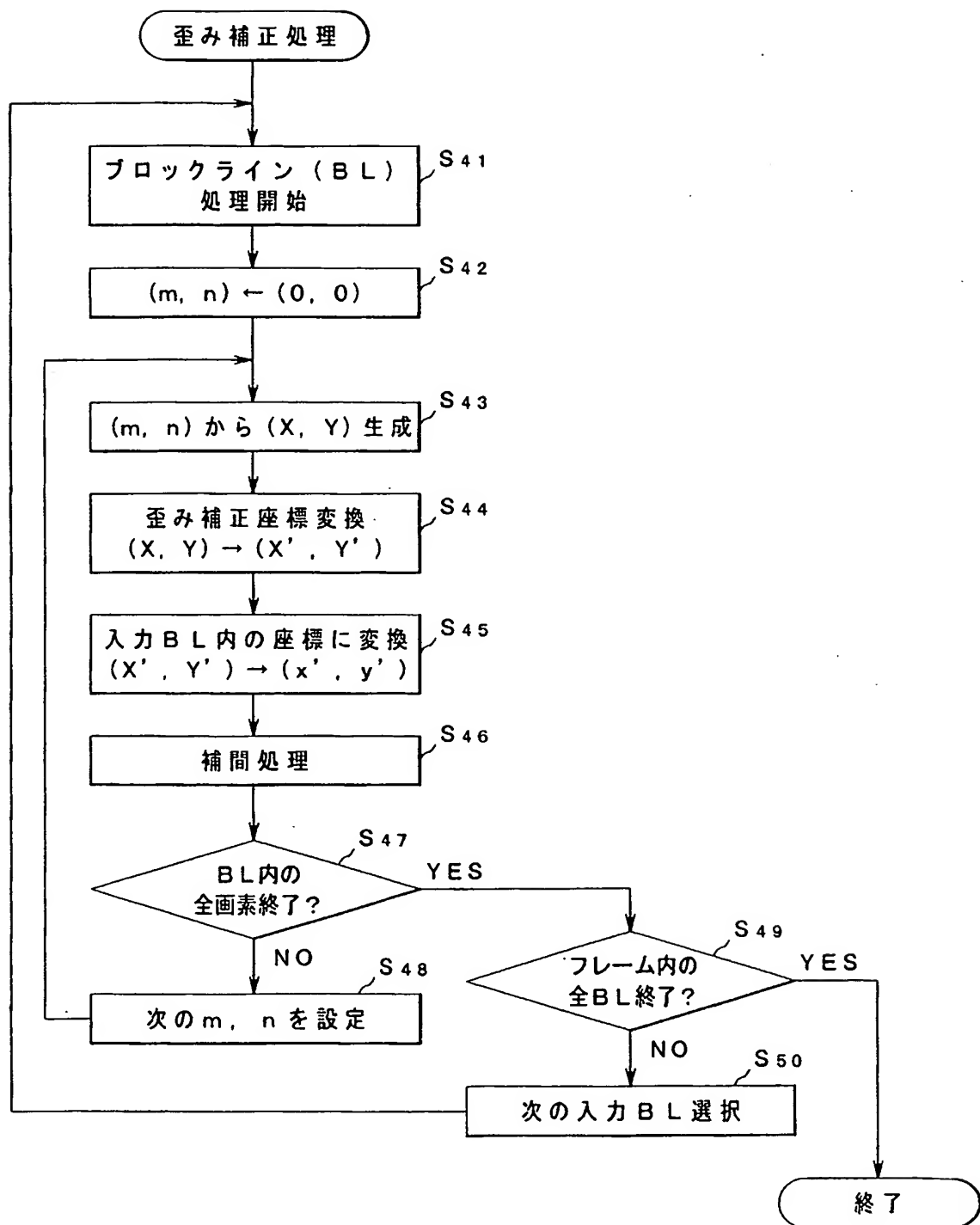


図 2 6

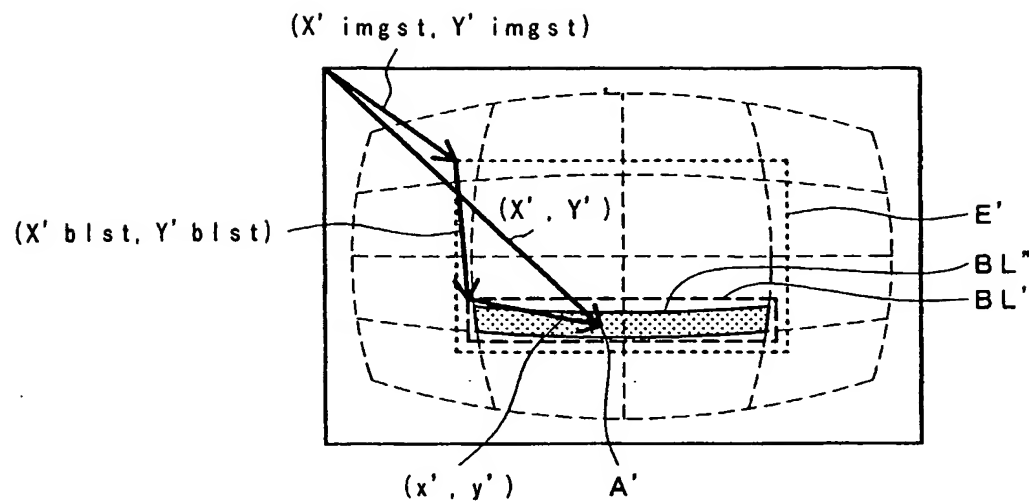


図 2 7

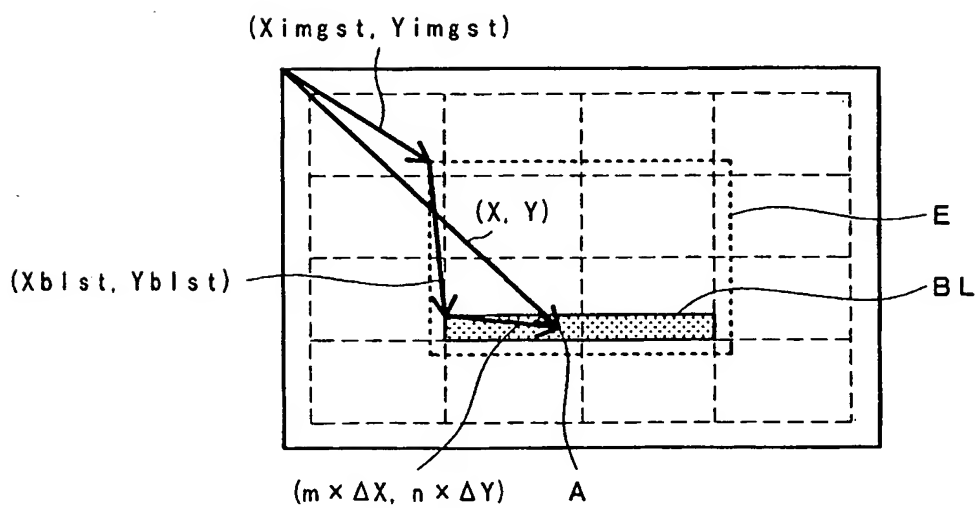


図 28

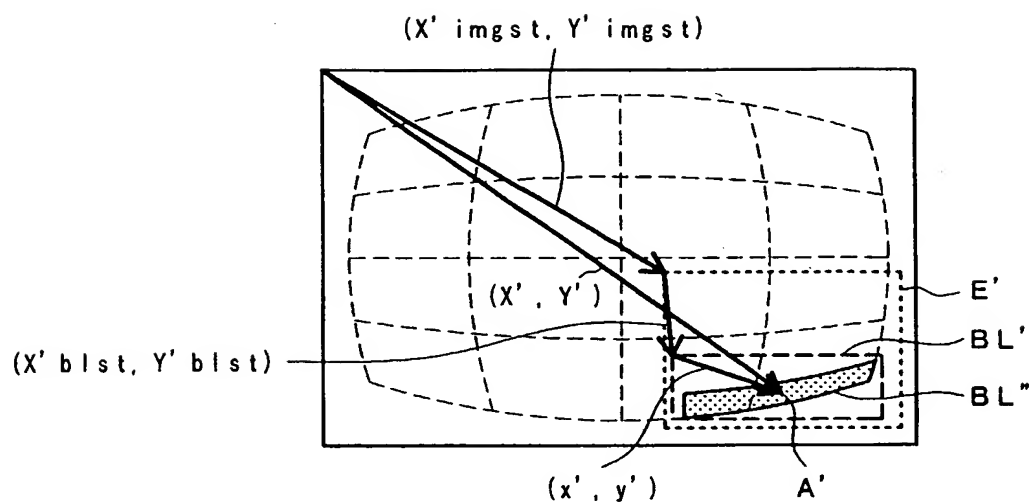


図 29

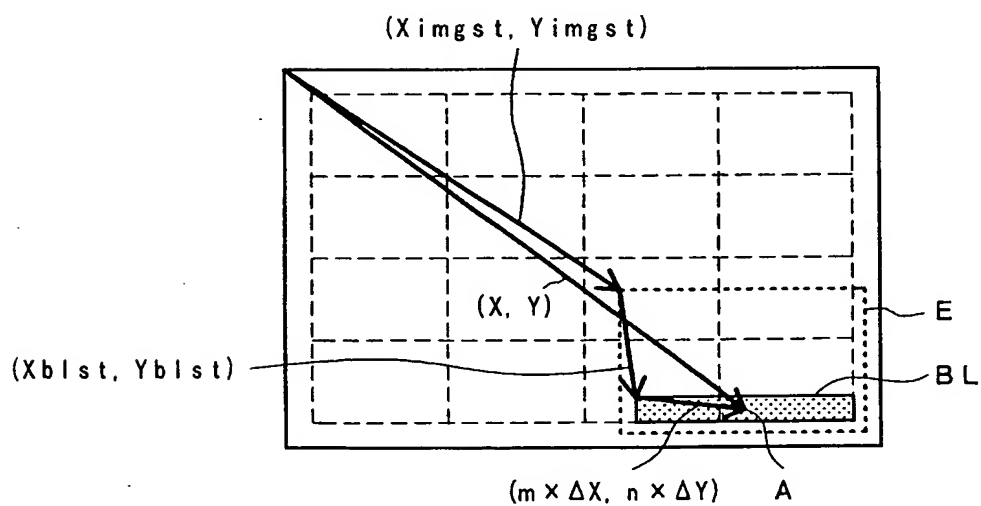


図 3 0

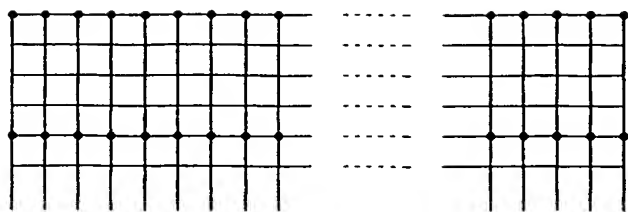


図 3 1

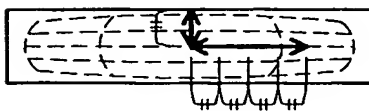


図 3 2

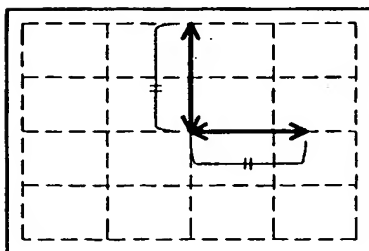


図 3 3

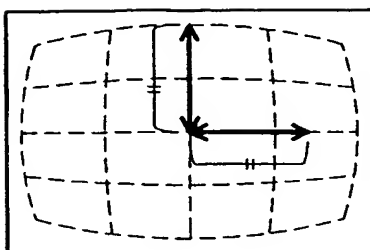


図 3 4

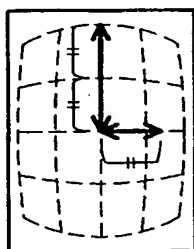


図 3 5

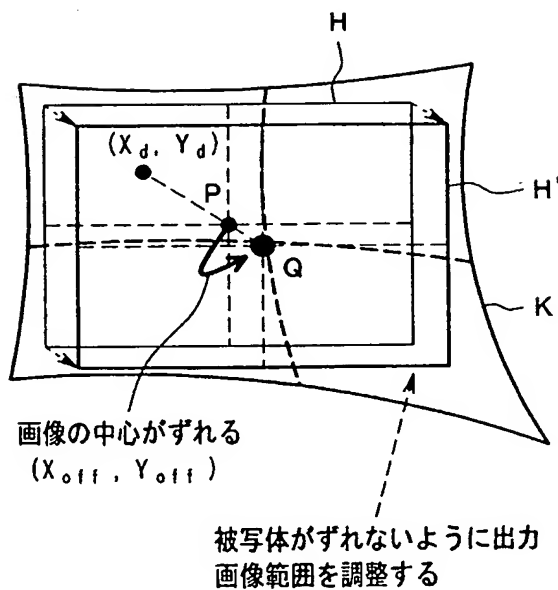
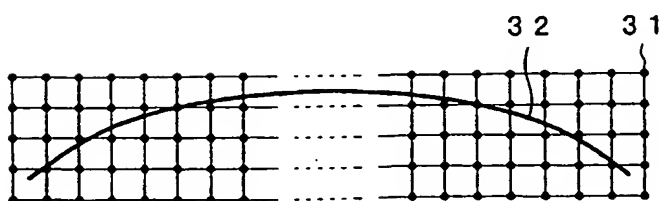


図 3 6



37

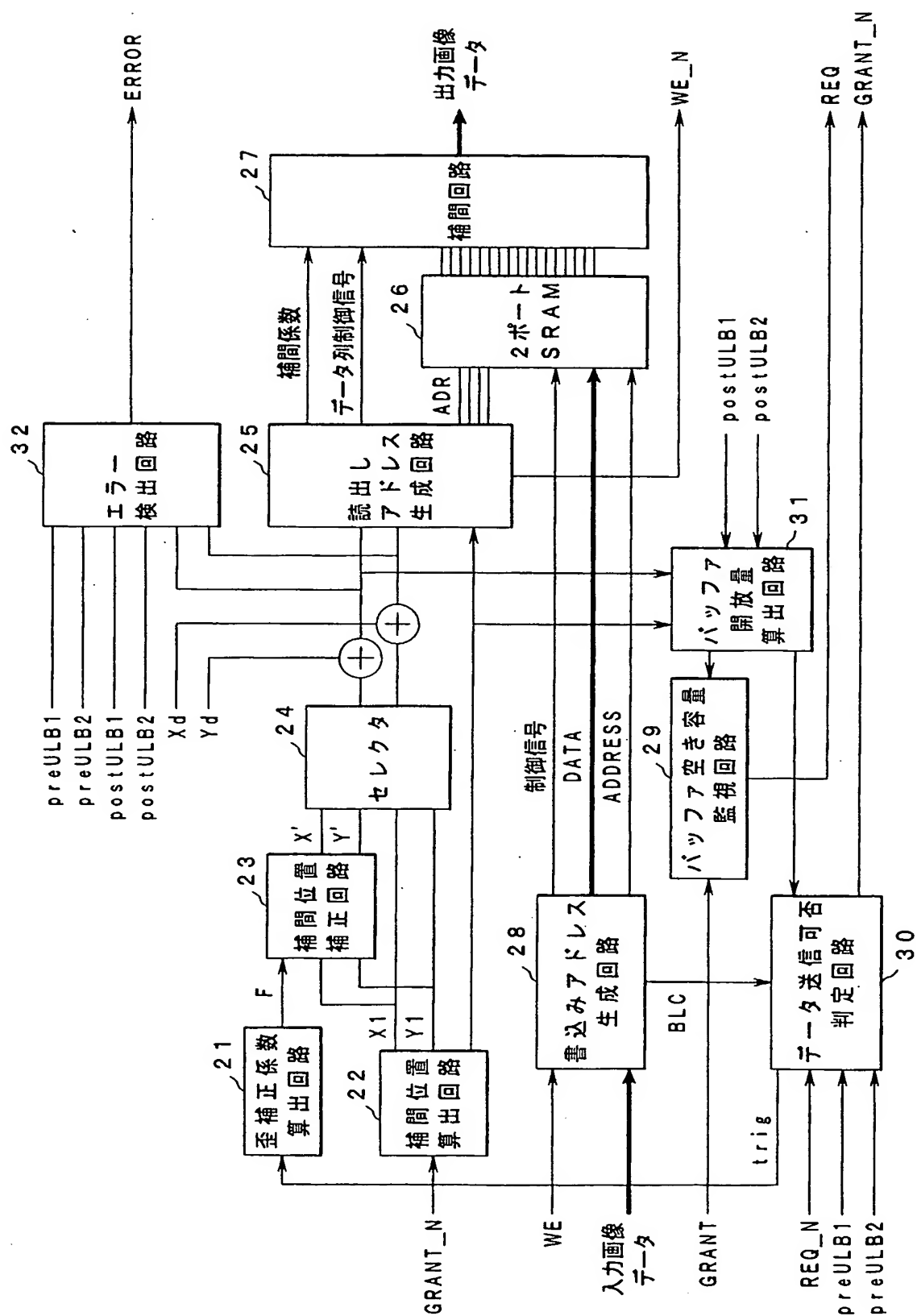


図 3 8

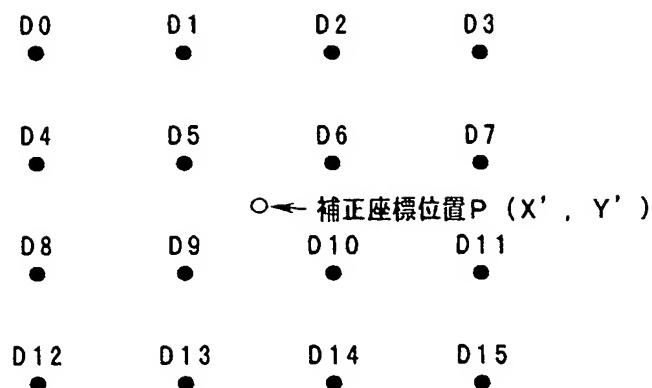


図 3 9

入力されるULの順 (N : 整数)

		4N+4 4N+12				4N+5 4N+13				4N+6 4N+14				4N+7 4N+15			
		4N	4N+8	4N+12	4N+16	4N+1	4N+9	4N+13	4N+17	4N+2	4N+10	4N+14	4N+18	4N+3	4N+11	4N+15	4N+19
アドレス	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
	2	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
	23	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92
		No. 0				No. 1				No. 2				No. 3			
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	2	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
	23	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93
		No. 4				No. 5				No. 6				No. 7			
	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	1	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
	2	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
	23	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94	94
		No. 8				No. 9				No. 10				No. 11			
	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	1	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
	2	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
	23	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95
		No. 12				No. 13				No. 14				No. 15			

図 4 0

1UL BL

1	5	9	13	A	B	C	D		
2	6	10	14	E	F	G	H		
3	7	11	15	I	J	K	L		
4	8	12	16	M	N	O	P		
a	b	c	d						
e	f	g	h						
i	j	k	l						
m	n	o	p						

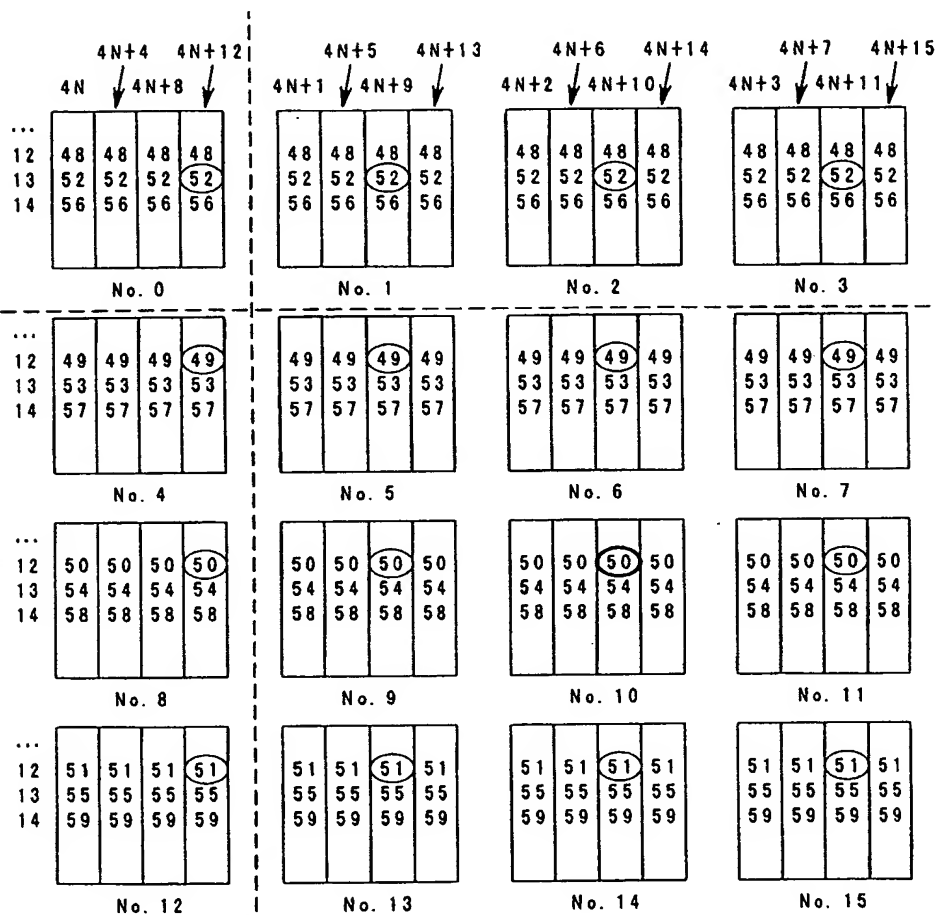
} BLの幅

図 4 1

1 a	A				5 b	B				9 c	C				13 d	D			
No. 0					No. 1					No. 2					No. 3				
2 e	E				6 f	F				10 g	G				14 h	H			
No. 4					No. 5					No. 6					No. 7				
3 i	I				7 j	J				11 k	K				15 l	L			
No. 8					No. 9					No. 10					No. 11				
4 m	M				8 n	N				12 o	O				16 p	P			
No. 12					No. 13					No. 14					No. 15				

图 4 2

ex. 補間位置 = $(4N + 10 + \alpha_0, 50 + \alpha_1)$ の場合 (α_* : 少数部)



各メモリへの読み出しアドレス

No. 5, 6, 7, 9, 10, 11, 13, 14, 15 : 60 (=24*2+12)

No. 4, 8, 12 : 84 (=24*3+12)

No. 1, 2, 3 : 61 (=24*2+13)

No. 0 : 85 (=24*3+13)

図 4 3

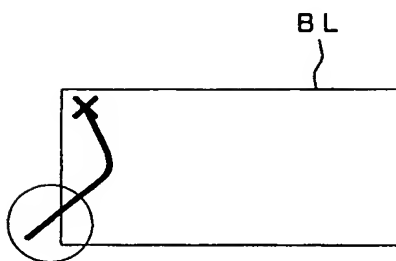


図 4 4

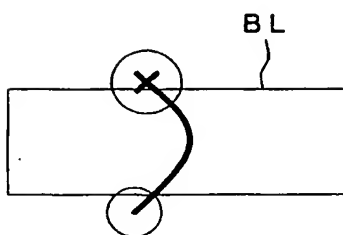


図 4 5

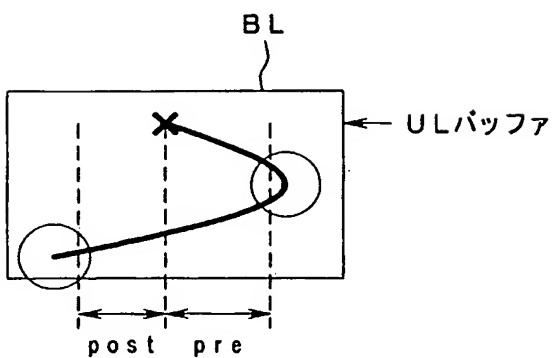


図 4 6

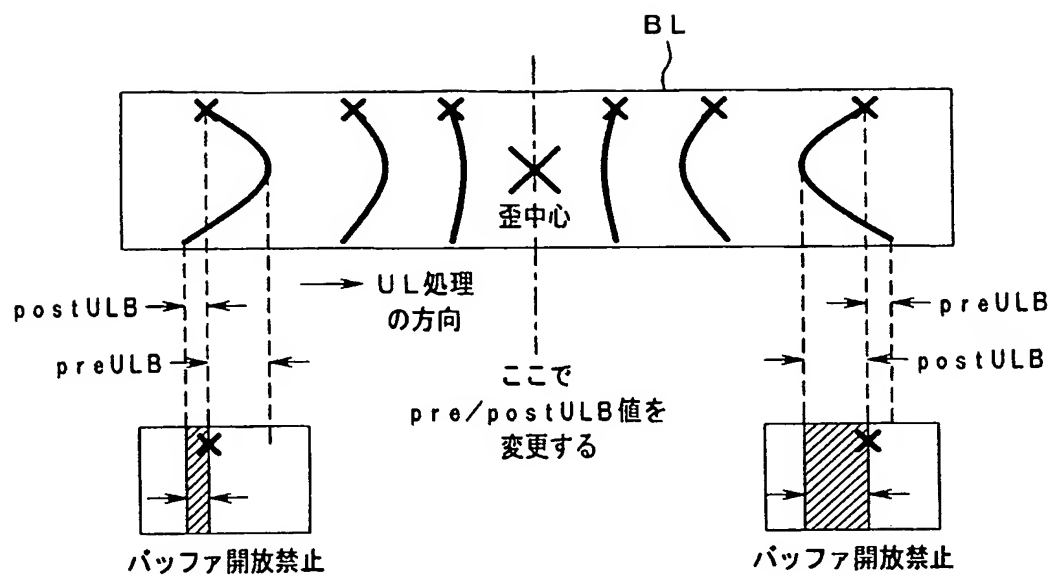


図 4 7

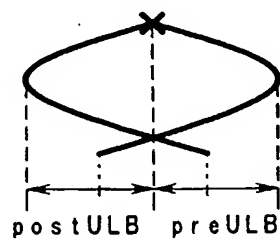


図 4 8

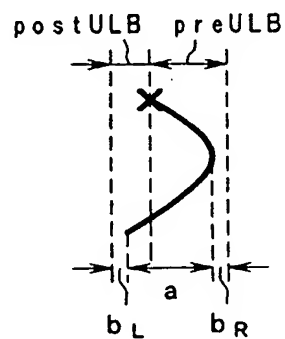


図 4 9

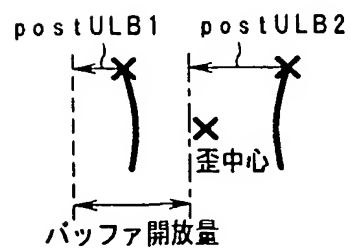


図 5 0

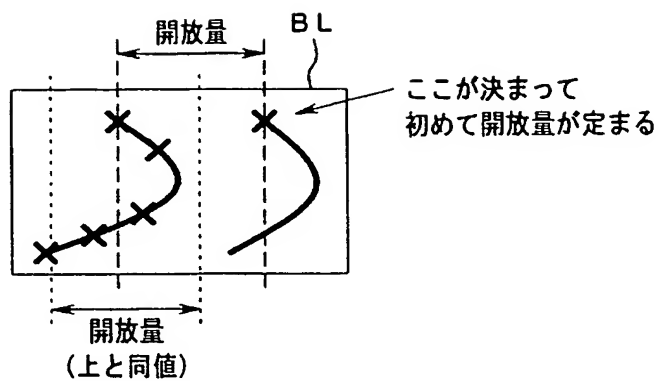


図 5 1

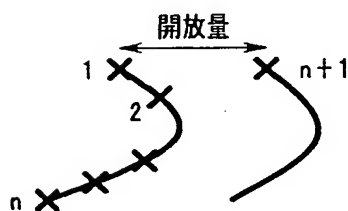


図 5 2

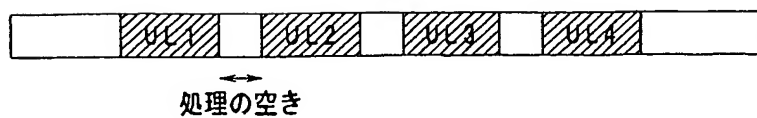


図 5 3

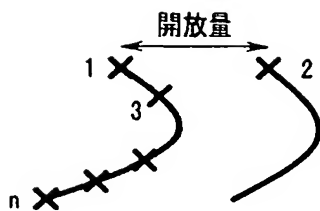


図 5 4



図 5 5

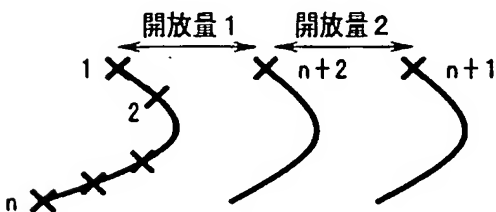


図 5 6

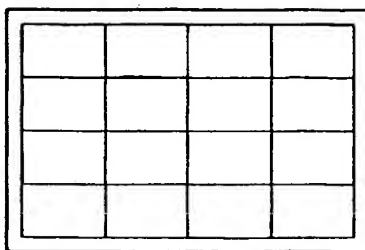


図 5 7

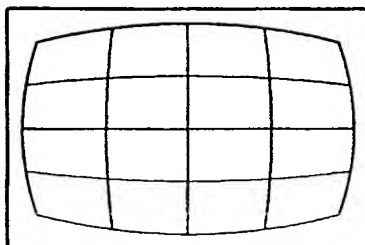


図 5 8

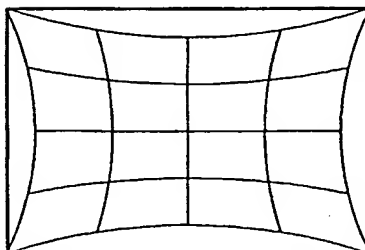


図 59

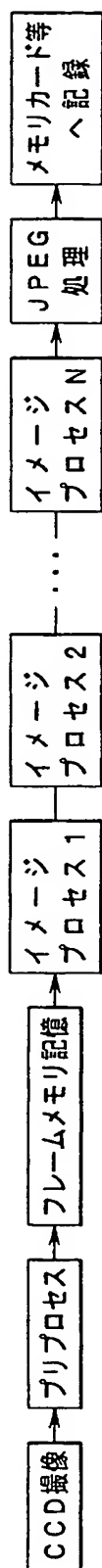


図 60

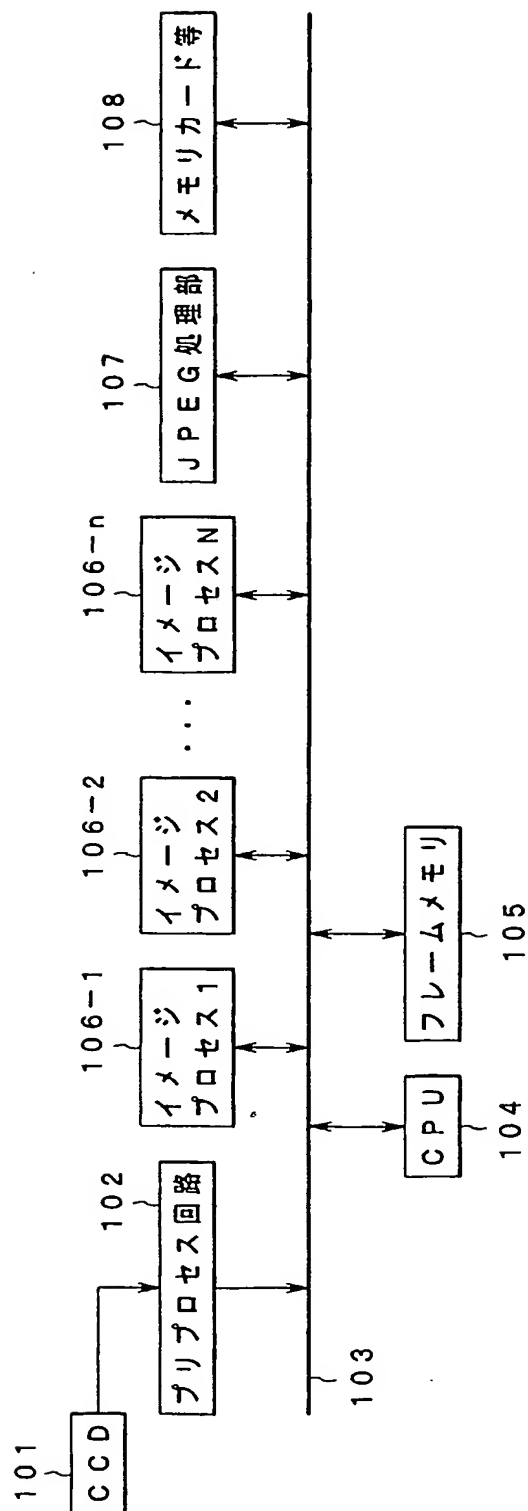


図 6 1

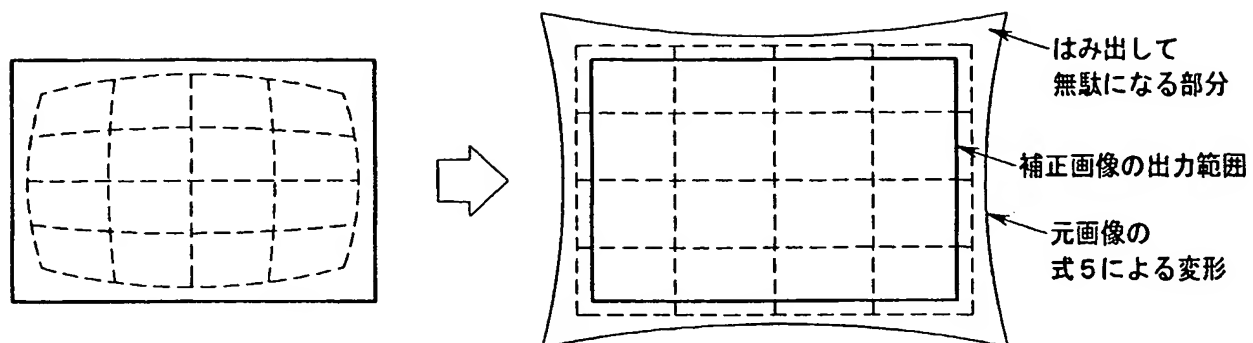


図 6 2

